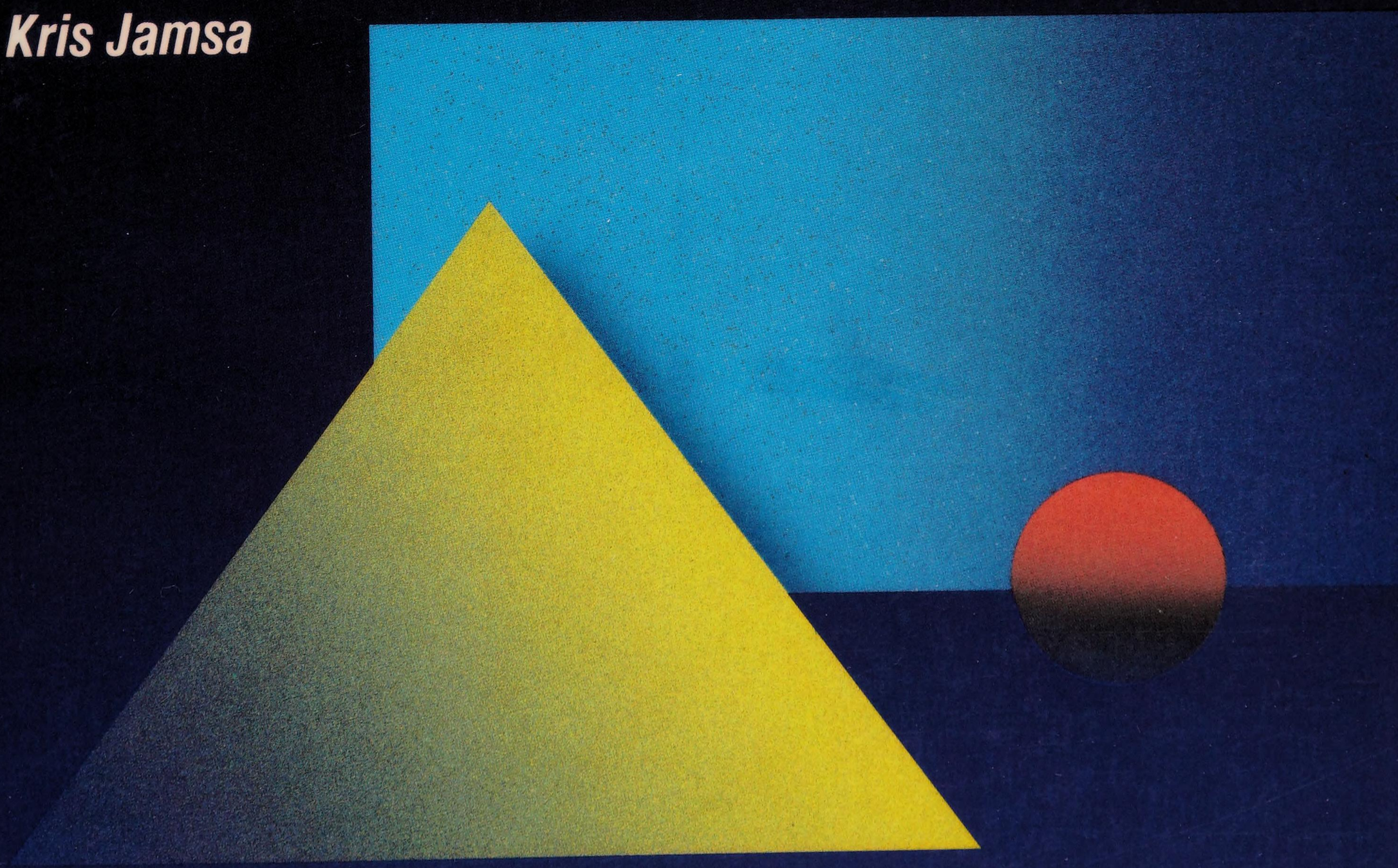
Osborne McGraw-Hill

DOS

SECRETS, SOLUTIONS, SHORTCUTS

Kris Jamsa



Covers All MS-DOS® & PC-DOS Versions Through 3.3

Benton K. Bovee
1773 House Box 210
Tinker Hill Lane
RR 5 Malvern, PA 19355

DOS:
Secrets, Solutions, Shortcuts

Kris Jamsa

Osborne **McGraw-Hill**
Berkeley, California

Osborne **McGraw-Hill**
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations and book distributors outside of the U.S.A., please write to Osborne **McGraw-Hill** at the above address.

A complete list of trademarks appears on page 763.

DOS: Secrets, Solutions, Shortcuts

Copyright © 1988 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

1234567890 DODO 898

ISBN 0-07-881471-5

Acquisitions Editor: Cindy Hudson
Copy Editor: Valerie Robbins
Word Processor: Bonnie Bozorg
Proofreaders: Kay Luthin, Juliette Anjos
Technical Illustration: Peter Hancik
Production Supervisor: Kevin Shafer

Information has been obtained by Osborne **McGraw-Hill** from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, Osborne **McGraw-Hill**, or others, Osborne **McGraw-Hill** does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from the use of such information.

Contents

Preface *xiii*

I *DOS Basics* *1*

1 *Getting Started with DOS* *3*

Starting Your System 8
Tips and Traps Summary 32

2 *Entering DOS Commands* *37*

Internal Versus External DOS Commands 40
The Default Disk Drive 46
Setting the System Date and Time 48
Tips and Traps Summary 52

3 *Handling Your Files* *55*

Looking at Directories 56
Naming Files 65
Using Wildcard Characters 66
Making Duplicate Copies of Your Files 70
Changing a File's Name 75
Deleting Files 78
Tips and Traps Summary 81

4 ***Handling Your Disks*** ***85***

Handling Your Floppy Disks 86
Preparing a Disk for Use by DOS 86
Naming Your Disks 95
Fixed Disks Versus Floppies 100
Preparing a Fixed Disk for Use by DOS 100
Examining the Current State of Your Disks 105
Contiguous Files 109
Tips and Traps Summary 115

5 ***Using System Devices*** ***119***

Creating Your Own Text Files 119
Using CON as an Ouptut Device 127
Looking at DOS Device Names 128
Using Your Printer 132
Configuring Your Devices 133
Controlling Your Screen Display 134
Controlling Your Printer Output 138
Configuring Your Serial Device 140
Tips and Traps Summary 142

II ***Putting DOS to Work for You*** ***145***

6 ***Getting the Most from Your Keyboard*** ***147***

Common Keyboard Layouts 147
Restarting DOS with SHIFT-PRTSC 149
Deleting Command-Line Characters with BACKSPACE 150
Canceling a Command Line with ESC 152
Terminating a Program with CTRL-BREAK 153
Suspending Output with CTRL-S 156
Printing the Contents of Your Current Screen with SHIFT- PRTSC 160
Echoing Text to Your Printer with CTRL-S 164
Using the DOS Function Keys 165
Editing Text in the Command Buffer 168
Enhancing Your Files with ALT 172
Tips and Traps Summary 178

7 ***Redirecting DOS Input and Output*** ***183***

Redirecting Command Output with > 183
Appending Redirected Output with > 187
Redirecting Command Input with < 189
Failing to Redirect MORE and SORT 199

Using the DOS Pipe Operator (|) 201
Placing Multiple Redirection Operators on One Line 206
Tips and Traps Summary 208

8 *Using the DOS PRINT Command* 211

Getting Started with PRINT 212
How PRINT Works 218
Removing Files from the Print Queue 219
PRINT Configuration Qualifiers 223
Tips and Traps Summary 230

9 *Customizing DOS* 233

Creating CONFIG.SYS 235
Enabling Extended CTRL-BREAK Checking 236
Reducing the Number of Disk I/O Operations 238
Defining Country-Specific Information 243
Setting Block Device Attributes 243
Supporting DOS Applications Prior to Version 2.0 245
Determining the Number of Files That DOS Can Open 247
Defining the Number of Logical Disk Drives Supported 248
Defining the Command-Line Interpreter 250
Creating a Hardware Interrupt Stack 251
Installing a Device Driver 253
Supporting External Floppy-Disk Drives 254

Installing a Virtual-Disk Drive in RAM 257
Having Fun with ANSI.SYS 260
Tips and Traps Summary 268

10 ***Utilizing the DOS Environment*** ***275***

Displaying the Default Environment 275
Assigning Environment Entries 278
How Many Entries Can the Environment Store? 280
Other Uses of the DOS Environment 283
Tips and Traps Summary 284

11 ***Using DOS Batch Files*** ***287***

A First Look at Batch Files 288
Terminating a Batch File 291
Suppressing Command-Name Display 294
Documenting Your Batch Files 298
Suspending Batch Processing 300
Batch-File Parameters 301
Tips and Traps Summary 308

12 ***Building Intelligent Batch Files*** ***311***

Using IF to Make Decisions 312
Using the DOS NOT Operator 315
Branching Within Your Batch Files Using GOTO 315
Testing for a Nonexistent Parameter 320
Applying a DOS Command to a Collection of Files 321
Using Multiple Batch Parameters with SHIFT 324
Named Parameters 328

Invoking Nested Batch Procedures 331
AUTOEXEC.BAT: A Special Batch File 334
Tips and Traps Summary 335

13 ***Using DOS Subdirectories*** ***339***

Getting Started with DOS Subdirectories 339
Using MKDIR to Create DOS Subdirectories 341
Selecting the Current Directory with CHDIR 346
Removing Subdirectories with RMDIR 350
Creating Multiple Levels of Subdirectories 353
Rules for Naming Your Directories 358
What Are . and ..? 358
Directory Listings of Files Contained in DOS Subdirectories 361
Manipulating Files Contained in DOS Subdirectories 362
Entering External Commands That Reside in DOS Subdirectories 364
Tips and Traps Summary 367

14 ***Advanced Use of Subdirectories*** ***371***

Taking a Look at Your Directory Structure 371
Using PATH to Locate DOS Command Files 376
Installing DOS on Your Fixed Disk 382
Creating a Data-File Search Path 384

Protecting Your Files with Unique Subdirectory Names 387
Upgrading DOS 389
Tips and Traps Summary 391

15 ***Using Logical Disk Drives and DOS Pretender Commands*** ***393***

Using Drives A and B in a Single-Floppy-Disk System 394
Making One Disk Drive Appear as Another 397
Making Your Disks Appear Larger than They Are 399
Abbreviating DOS Subdirectory Names 402
A Word of Warning 405
Tips and Traps Summary 406

16 ***Getting the Most from the DOS File-Manipulation Commands*** ***409***

Revisiting the FIND Command 410
Recovering a Damaged File or Disk 412
Setting and Displaying File Attributes 416
Comparing the Contents of Two Files 422
Using FC to Compare Files 425
Performing Selective File Replacements 430
Tips and Traps Summary 436

17 ***COPY Versus XCOPY*** ***439***

Exploring the COPY Command in Detail 439
Working with the DOS XCOPY Command 445

When to Use XCOPY Instead of COPY 454
Tips and Traps Summary 456

18 ***Getting the Most from DOS System Commands*** ***459***

Verifying your Disk-Output Operations 459
Changing the DOS Prompt 461
What Is CTTY? 467
Improving Disk-I/O Performance with FASTOPEN 468
Using the DOS EXIT Command 470
Using DISKCOMP to Compare Two Floppy Disks 473
Enhancing Character Display with GRAFTABL 475
File Sharing in a Local-Area Network 477
Tips and Traps Summary 480

19 ***International DOS Support*** ***485***

Using the Correct Character Set 485
Selecting a Keyboard Template 486
What Is a Symbol Set? 490
What You Need to Get Started 491
Multiple-Language Users 492
Using Mode for Codepage Support 496
Using the DOS SELECT Command 499
Tips and Traps Summary 500

20 ***Using BACKUP and RESTORE*** ***503***

Floppy-Disk-Based Systems 504
When to Use BACKUP 505
Getting Started with BACKUP 506
Using the RESTORE Command 513
Labeling Your Disks 515
Developing a Backup Policy 516
What Is a Log File? 520
Batch-File Processing 521
Correcting Disk Fragmentation 522
Upgrading Your Fixed Disk to a New Version of DOS 525
Tips and Traps Summary 527

21 ***Using DEBUG to Enhance Your Batch-File Processing*** ***531***

Creating a Program with DEBUG 532
Obtaining the Current Disk Drive 536
Determining the State of Extended CTRL-BREAK Processing 537
Determining the State of Disk Verification 539
Getting a User Response 540
Getting a Function-Key Response 541
Tips and Traps Summary 544

A ***ASCII Codes*** ***545***

B ***Command Reference*** ***549***

Index ***765***

Preface

In 1981, IBM announced the IBM PC, and with the PC came DOS, the operating system for the PC and PC compatibles. DOS, as you will learn, is the overseer of your computer's resources (disks, printers, and so on). In addition, DOS allows your computer to run other programs. Therefore, you must have DOS running on your PC before the computer can perform useful work.

Is This Book for You?

This book is about DOS, both MS-DOS and PC-DOS. By following the chapters in this book as they are presented, novices will become proficient with DOS in no time. Intermediate users will learn the secrets and shortcuts to save time and get the most from DOS that advanced users have found. Advanced DOS users will find many techniques not discussed in the DOS documentation or other DOS books.

Each chapter presents unique pieces of information to DOS users at all levels. By following through the examples in this book, you will learn the secrets, solutions, and shortcuts that I have found by using DOS every day since 1981. If you spend a minute or two reading the tips and traps found in some of the chapters, you will know this book is for you.

A Note to Experienced DOS Users

The first part of this book gets new users familiar with the basics of DOS. Since many readers will be new to DOS, I have provided a solid foundation before examining the advanced material presented in Part II.

If you have been using DOS for some time, skim the tips and traps in the early chapters, then move on to Part II. Have fun!

Disk Package

One of the greatest frustrations when working with DOS is forgetting a command or command format. Therefore, this book provides a Command Reference section in Appendix B that summarizes each DOS command. However, if you are working at your computer and you do not want to have to reach for your book, a "DOS HELP" package providing a discussion of each DOS command, as well as the tips and traps found in this book, is available on disk.

The disk package costs $17.95 plus $2.00 shipping and handling ($5.00 on foreign orders).

Please send me the disk package that accompanies *DOS Secrets, Solutions, Shortcuts*. My payment for $19.95 ($17.95 plus $2.00 for shipping and handling or $5.00 for foreign orders) is enclosed.

Name _____

Address _____

City _____ **State** _____ **Zip** _____

Kris Jamsa Software, Inc. Box 26031 Las Vegas, Nevada 89126

I

DOS Basics

Over 15 million people currently use DOS. Since each user possesses a different level of DOS experience, the first part of this book lays a foundation that will get everyone up to the same level of expertise. You must thoroughly understand the fundamental DOS concepts before you can exploit the concepts that will be discussed throughout the remainder of the book.

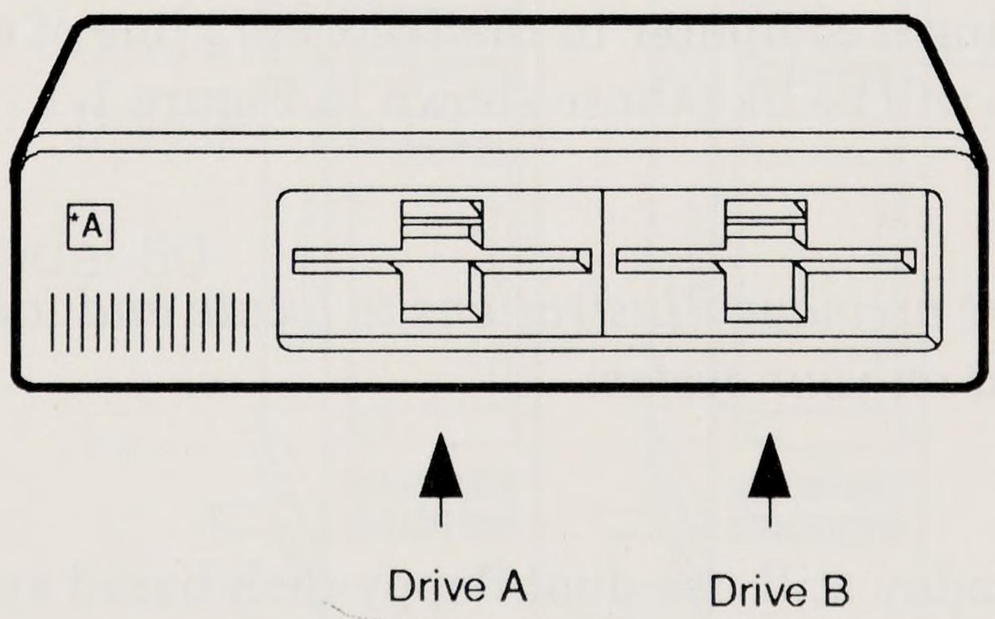
If you have been working with DOS for some time, the information in this first part provides an easy way for experienced users to review important facts that you may have forgotten. As you read these early chapters, keep in mind that much of the discussion is meant for new DOS users.

1

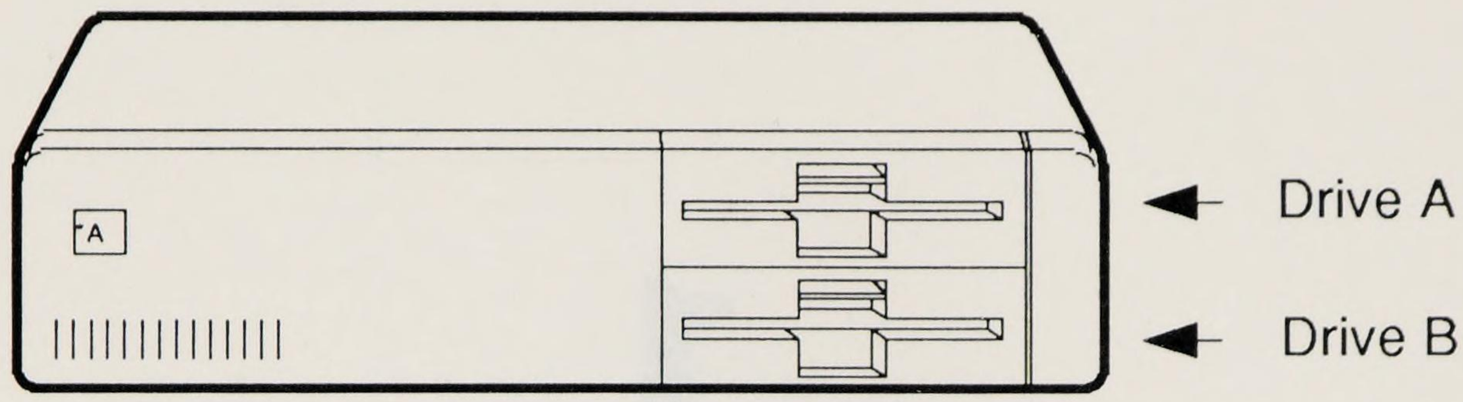
Getting Started with DOS

To ensure that everyone is up to speed with DOS and has a strong enough foundation to understand and appreciate the concepts that this book will examine, the best place to begin is with starting DOS.

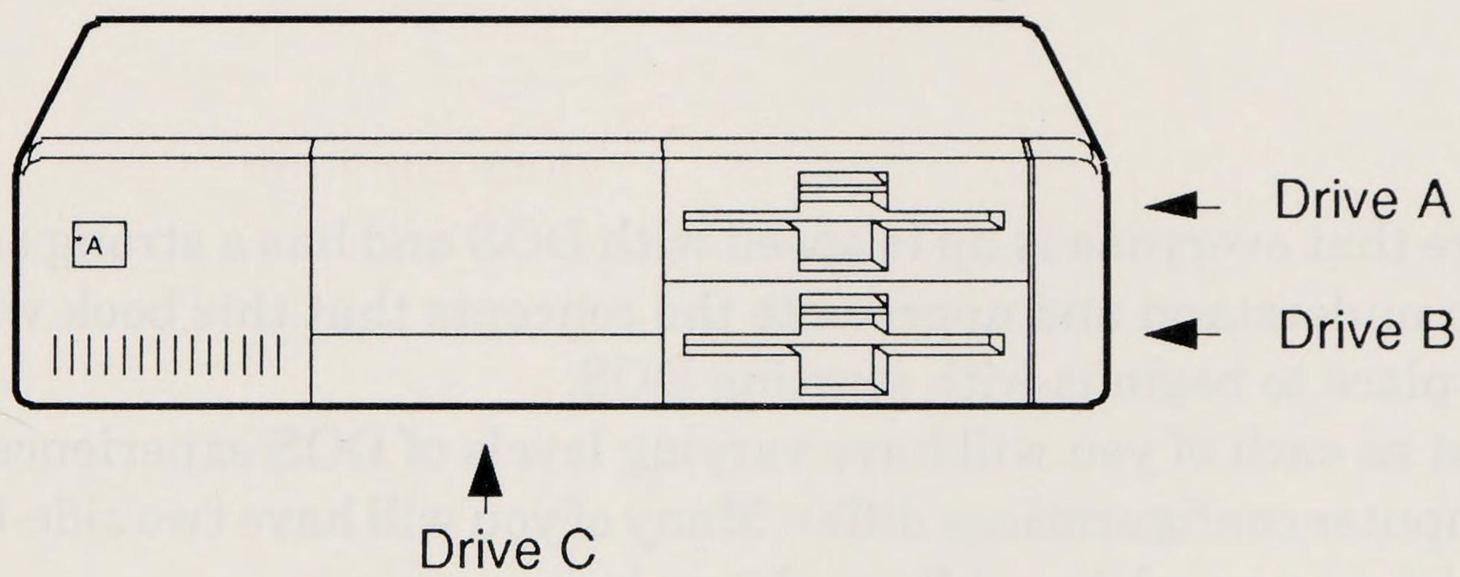
Just as each of you will have varying levels of DOS experience, so too will your computer configurations differ. Many of you will have two side-by-side floppy disk drives named A and B, as shown here:



Some of you may have half-height floppy disk drives also named A and B, as shown here:



Lastly, many of you will have a fast disk drive that is capable of storing many times more information than a floppy disk--a fixed disk. Your fixed disk will be named drive C, as shown here:



If you are using a computer in the IBM PS/2 line of computers, your disk drive assignments will be like those shown in Figure 1-1.

Tip: *Use the previous illustrations to locate and identify the disk drives present on your system.*

Most DOS users today still use dual-floppy-disk-based systems. The first few chapters of this book assume that you are starting DOS from a floppy-disk drive. Those of you with fixed disks might want to pull out your original DOS disks.

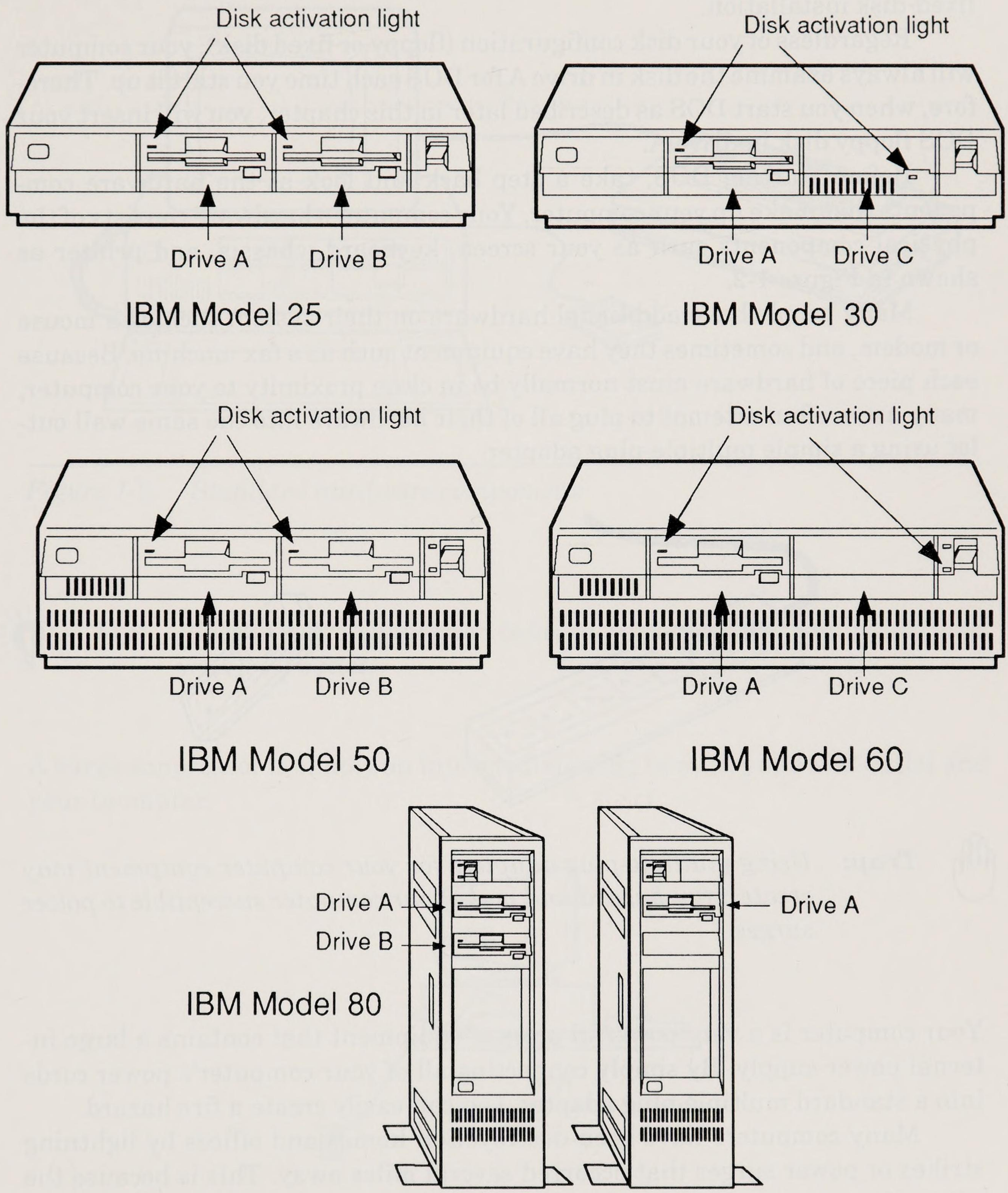


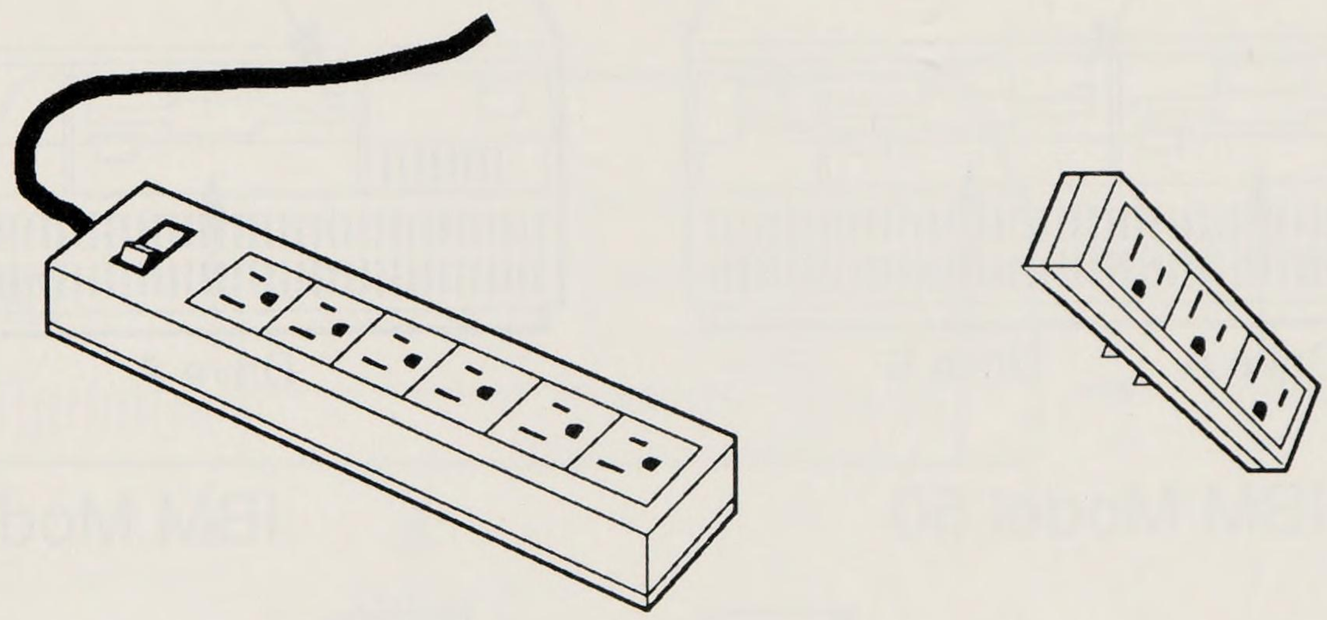
Figure 1-1. PS/2 models

You can use them throughout the early chapters before examining the DOS fixed-disk installation.

Regardless of your disk configuration (floppy or fixed disk), your computer will always examine the disk in drive A for DOS each time you start it up. Therefore, when you start DOS as described later in this chapter, you will insert your DOS floppy disk in drive A.

Before starting DOS, take a step back and look at the hardware components that make up your computer. Your computer's hardware consists of the physical components such as your screen, keyboard, chassis, and printer as shown in Figure 1-2.

Many people have additional hardware on their system such as a mouse or modem, and sometimes they have equipment such as a fax machine. Because each piece of hardware must normally be in close proximity to your computer, many users often attempt to plug all of their hardware into the same wall outlet using a simple multiple-plug adapter.



Trap: *Using multiple-plug adapters for your computer equipment may create a fire hazard and make your computer susceptible to power surges.*

Your computer is a very powerful piece of equipment that contains a large internal power supply. By simply connecting all of your computer's power cords into a standard multiple-plug adapter, you can easily create a fire hazard.

Many computers have been destroyed in homes and offices by lightning strikes or power surges that occurred several miles away. This is because the surge of power traveled over the power lines into the computer, burning up its internal circuitry. In most cases, such damage could have been easily prevented if the user had purchased a surge suppressor.

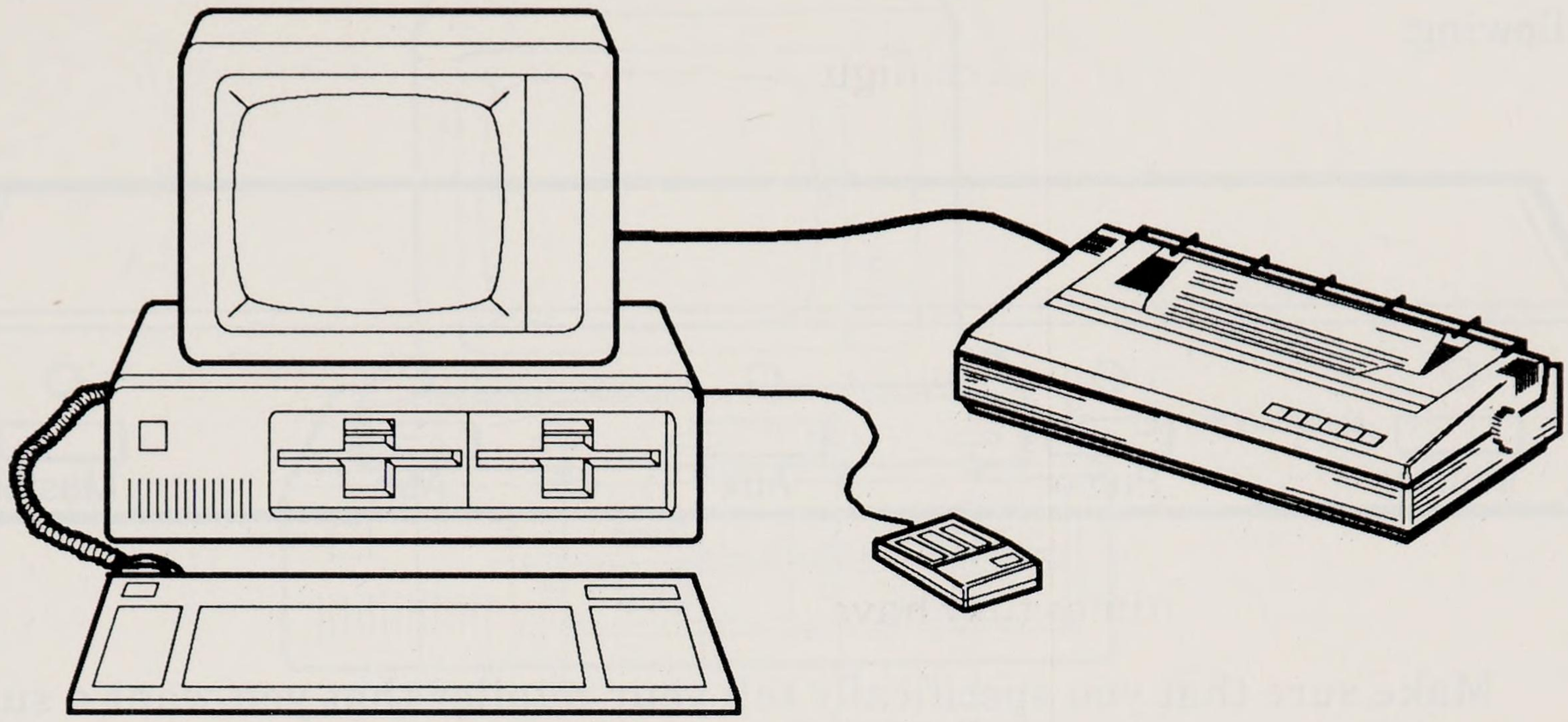
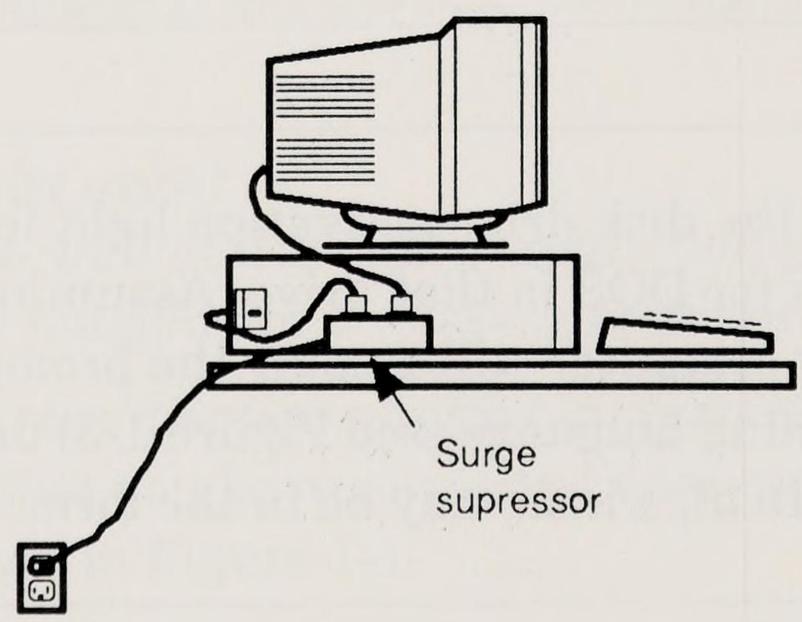


Figure 1-2. Standard hardware components

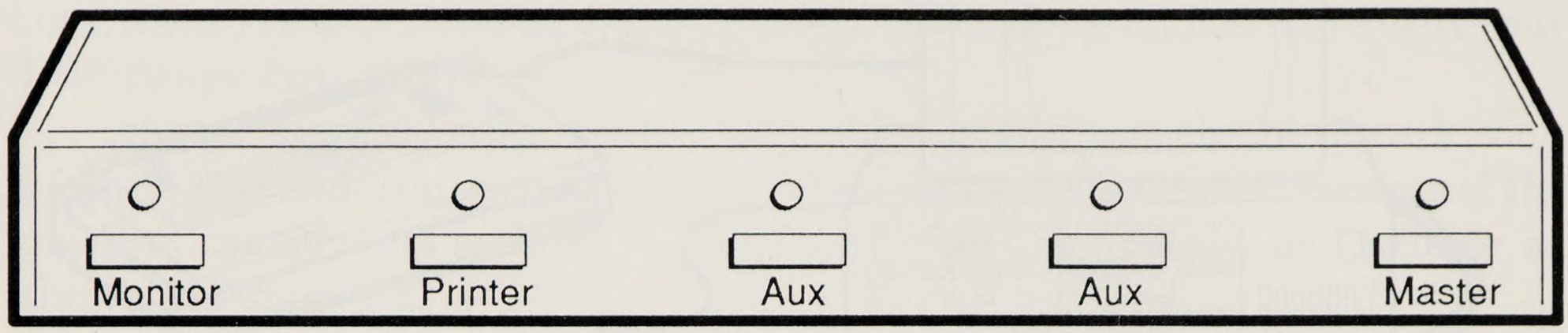
Tip: *Place a surge suppressor between your computer equipment and your wall outlet.*

A surge suppressor is simply an intermediate plug beween your wall outlet and your computer.



Should a large surge of power come down the power lines, the supressor will prevent the power from reaching your computer, in turn protecting your hardware investment. You can purchase surge suppressors that allow you to

easily power your computer equipment on and off individually as shown in the following.



Make sure that you specifically tell your retailer that you want a surge suppressor.

Starting Your System

There are a few steps that you can take to ensure that your computer is working prior to loading DOS from disk into your computer's memory. With your computer's power off, remove any floppy disks that may be in your computer's drives. Next, turn on power to your screen display and computer.

At this point, your computer should begin its power-on sequence, displaying on your monitor the amount of memory it successfully locates.

```
00640 KB OK
```

Next, you will see the disk drive activation light for drive A illuminate as your computer searches for DOS in that drive. Assuming that you don't have a fixed disk present, your computer will display the prompt for a built-in version of the BASIC programming language (see Figure 1-3) or an error message that states it could not find DOS, which may be in the form of

```
Disk Error Drive A
```

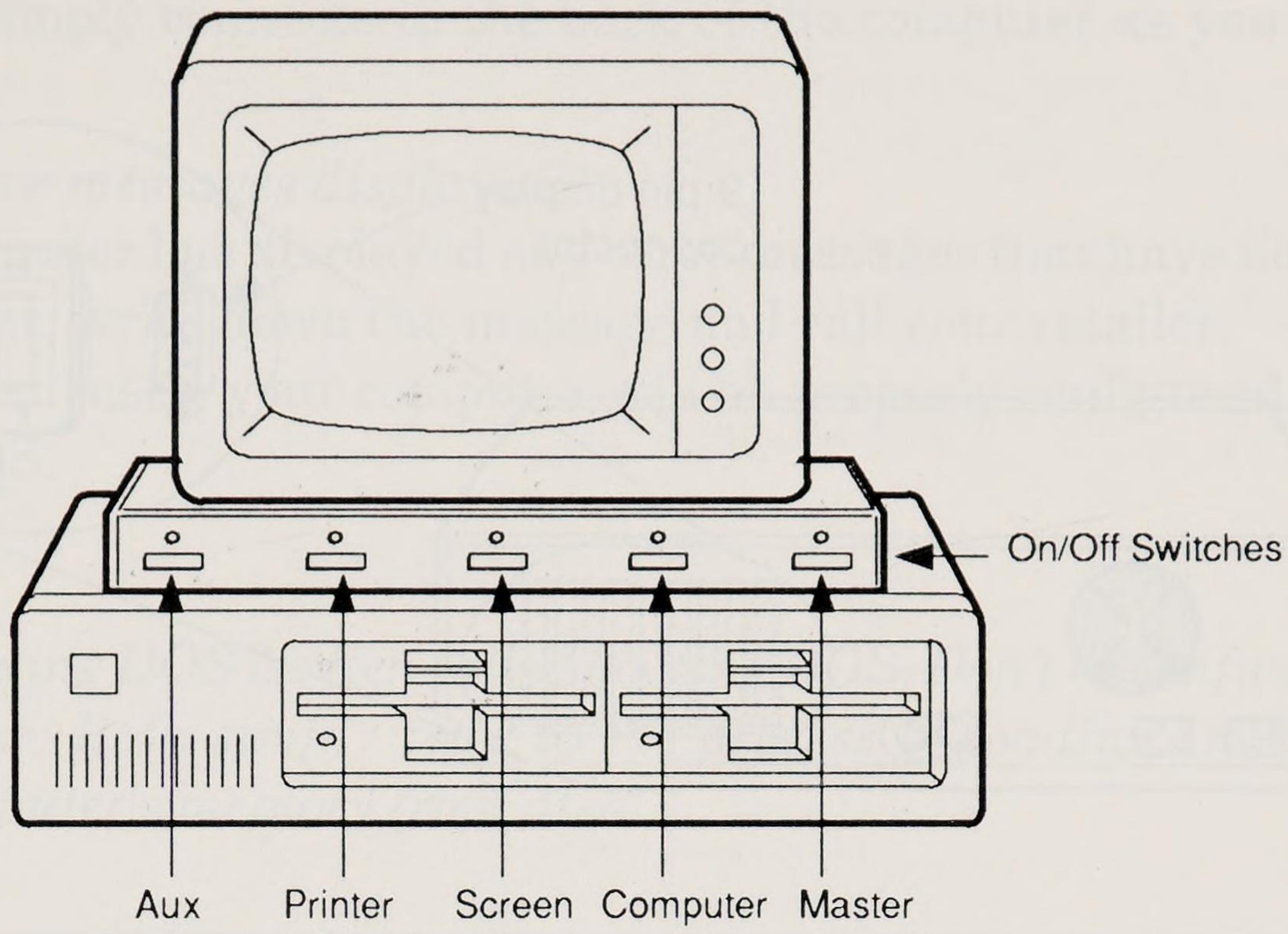


Figure 1-3. Distribution box that sits under the monitor

If this occurs, your computer is working and is ready to start DOS. If this doesn't occur, you may have a hardware problem or your system may not be correctly configured.

Tip: *If your computer is not working correctly, ask yourself the following questions.*

Does the wall outlet work?
Double-check the wall outlet that you are using for your computer and screen display by plugging in a working appliance.

Is the monitor (screen display) plugged in, turned on, and working?
If so, make sure that you have connected your monitor to the back of your computer as shown in Figure 1-4.

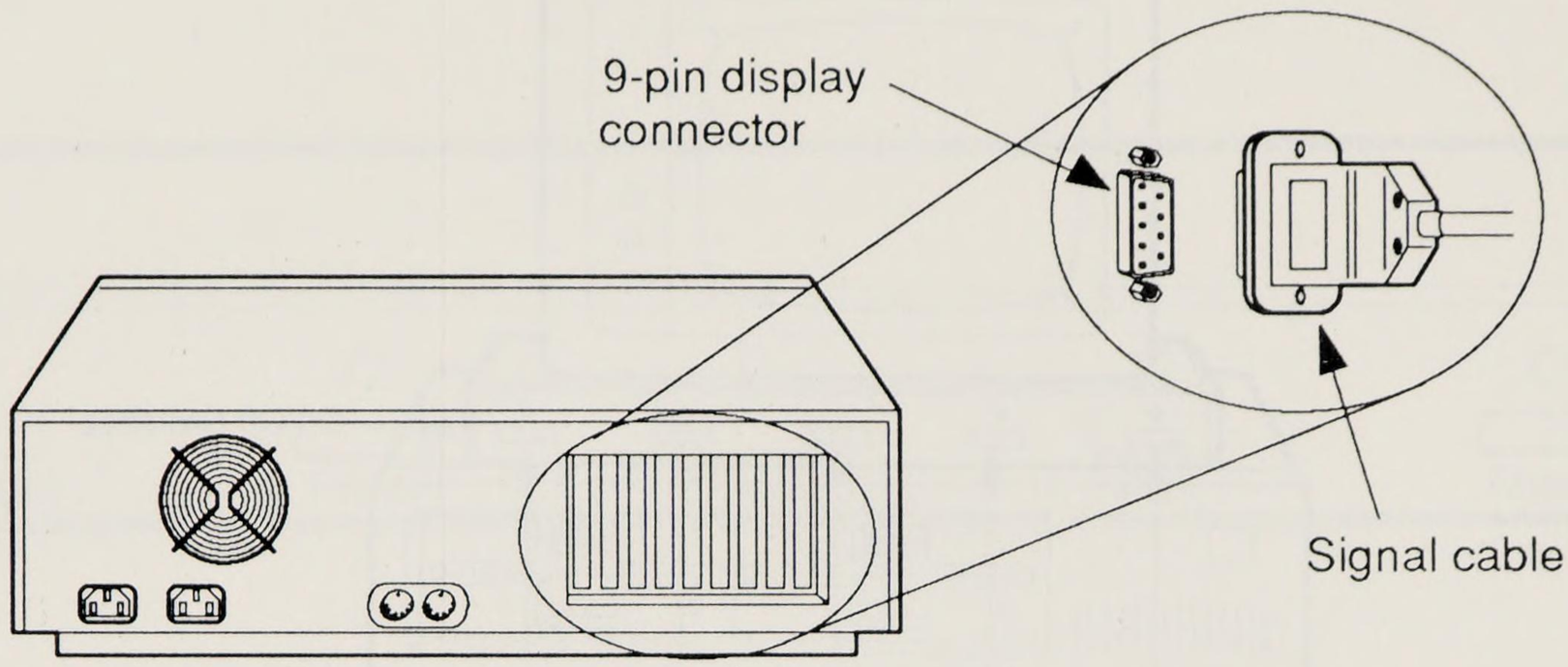


Figure 1-4. Monitor connection to computer

Can you hear the computer's fan once the computer is powered on?
To keep its internal hardware boards cool, your computer has a built-in fan that circulates cool air into it. This fan is noisy enough that it can be easily heard. If you can't hear your computer's fan, double-check your power plugs. If everything is correctly powered, you may have a power supply problem within your computer. Call your retailer.

Does your computer display the amount of memory it locates?
Most personal computers display the amount of working memory they find during the power on self test. If your computer doesn't, notify your retailer.

Does the disk drive activation light illuminate for drive A?
As stated previously, your computer always searches for DOS in drive A. Thus, if the computer passes its power-on self test, it will begin DOS start-up procedures by looking for DOS in drive A. If your computer does not pass the self test, it won't get to the point of looking for DOS on drive A.

Is the keyboard properly connected?
For your computer to function, you need only connect your screen display and keyboard to your computer's chassis. On most computer systems, the

keyboard simply connects to the back of the computer as you can see in Figure 1-5.

Are any error messages displayed?
If your computer has displayed any error messages that have not been discussed so far, write down the message and call your retailer.

In most cases, your computer will be properly configured and ready to start DOS.

Tip: *Starting DOS is often called booting DOS. Don't be confused by these terms. Both simply refer to the process of loading DOS into your computer's memory from disk.*

As mentioned earlier, your printer, monitor, keyboard, and computer's chassis comprise your computer's hardware. Your computer's software, on the other hand, consists of the programs that your computer runs. A word processor, spreadsheet program, accounting package, and even DOS are software. Software tells your computer's hardware which tasks to perform. Both hardware and software are worthless without each other.

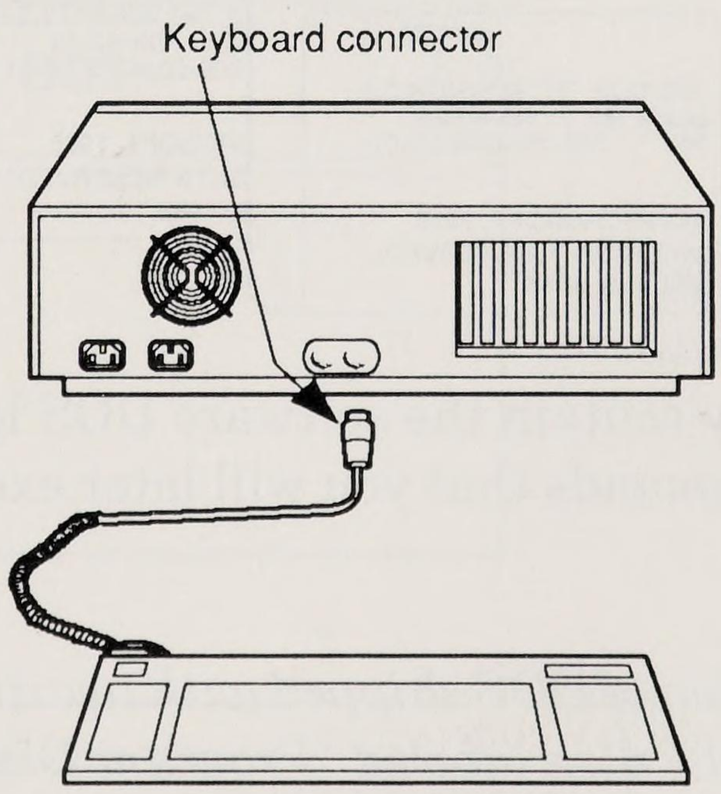
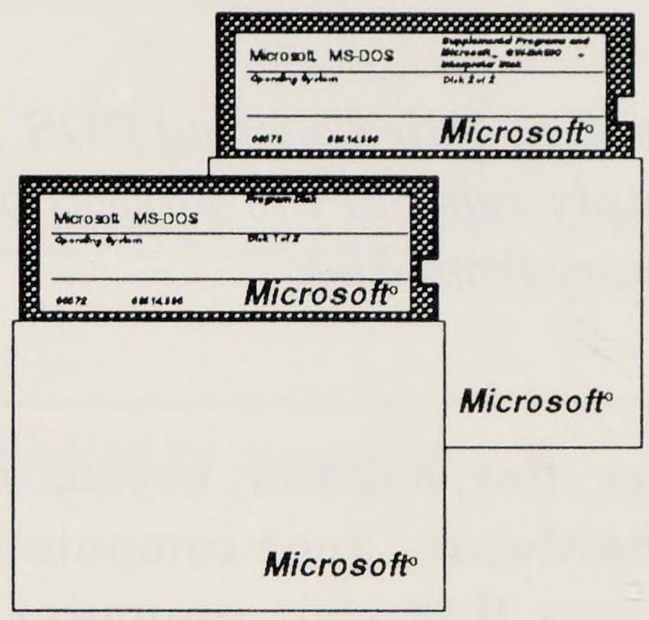


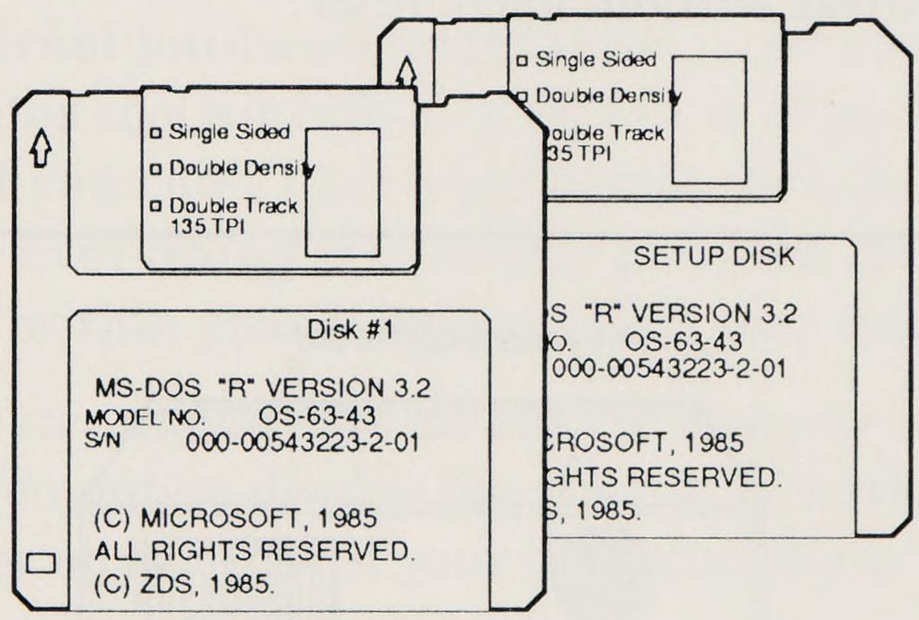
Figure 1-5. Keyboard connection to computer

DOS is a special software package called an *operating system* (DOS stands for Disk Operating System). In general, an operating system oversees your computer's *resources* (disks, printer, and so on) and allows you to run other programs. DOS must be the first program loaded into your computer's memory. Once present in your computer, DOS allows you to run other programs.

Take out your original DOS disks. Depending upon your computer's disk drive type, you will have either two 5 1/4-inch floppy disks labeled

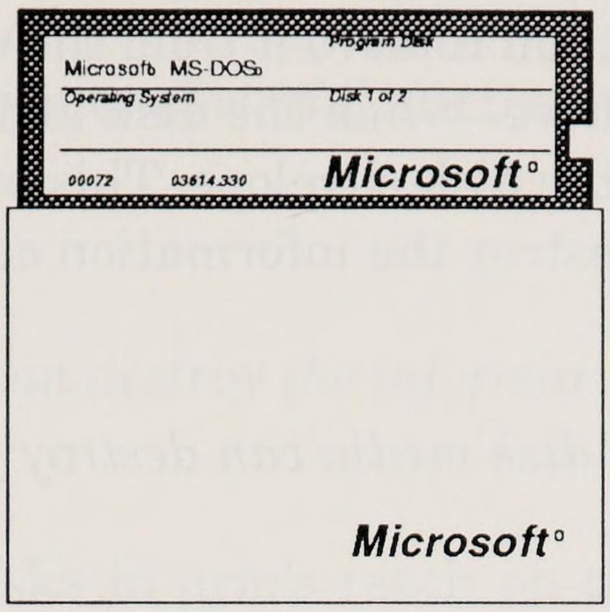


or one or more 3 1/2-inch microfloppy disks.



These disks simply contain the software DOS loads into memory when it starts and the DOS commands that you will later execute at the DOS prompt.

Tip: *If your version of DOS shipped with two disks, you will start (or boot) DOS from the disk labeled "Program Disk."*



Before starting DOS, there are some precautions you should know about for taking care of your disks.

Tip: *Always keep your disks in their disk envelopes when you are not using them.*

This tip is for 5 1/4-inch disks, which always have a protective disk envelope (see Figure 1-6).

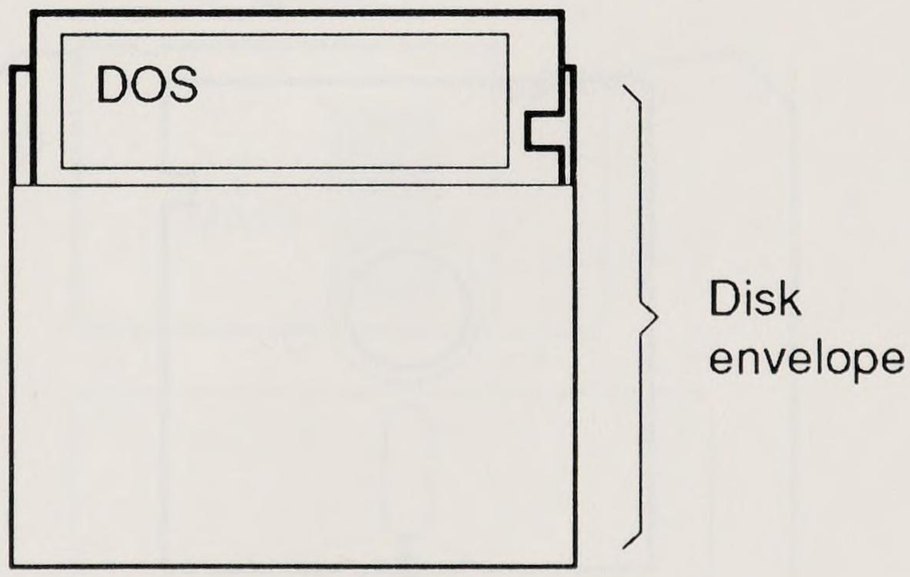


Figure 1-6. Protective disk envelope

When you use the disk, you remove it from the envelope in order to insert it into the computer's disk drive. When the disk is not in use, make sure that you place the disk back into the disk envelope. This protects the disk from dust, smoke, and spills that can destroy the information on the disk.

Trap: *Touching your disk media can destroy your disk's contents.*

If you remove a 5 1/4-inch disk from its protective envelope, you will see the actual *disk storage media* through the disk's read/write opening as shown in Figure 1-7. Likewise, if you slide the shutter of a microfloppy disk to the left, you will expose its storage media (see Figure 1-8).

The information stored on your disk is magnetized to the media. If you touch the disk storage media, you can destroy the information simply with a fingerprint or a small scratch.

Trap: *Magnetic and powerful electronic devices can destroy the information on your disks.*

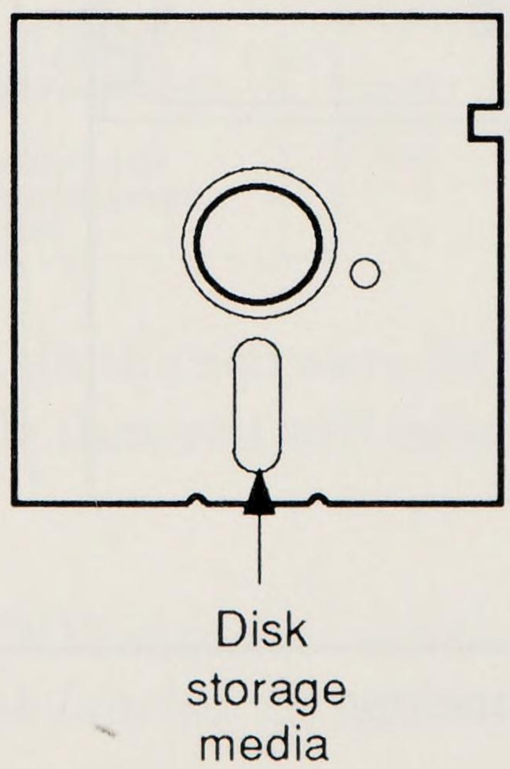


Figure 1-7. Disk storage media

As just stated, your disk stores information by magnetizing information. If you place your disk near a magnet or powerful electronic device, you can destroy the information on your disk.

Trap: *Your phone can destroy the information stored on your disks.*

Most people keep their disks in arm's reach on their desk, along with their telephone. Unfortunately, it is possible for your phone to destroy your disk's contents when it rings! Therefore, at a minimum, keep your disks several feet away from your phone.

Tip: *Always make a duplicate copy of each of your disks.*

Regardless of whether you are using 5 1/4-inch or 3 1/2-inch disks, you must always make a *backup copy* of each of them. Later in this chapter you will see how to do so with the DOS DISKCOPY command.

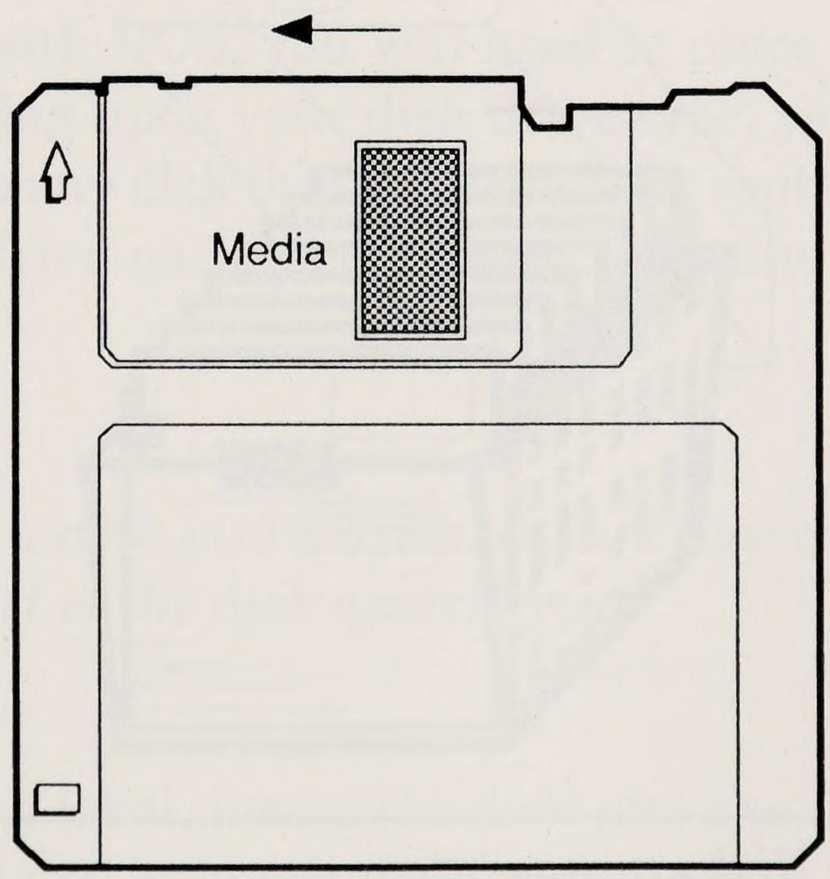


Figure 1-8. Disk media on microfloppy

Trap: *Smoking near your computer or disks can destroy your disks.*

As stated, your floppy disks are very sensitive. If smoke particles collect on the disk storage media, the particles can scratch the disk, destroying the information that it contained.

Tip: *Place your disks in a solid disk storage box for safety.*

When you are not using your disks, place them safely into a disk storage box as shown in Figure 1-9. This box will protect your disks from dust, smoke, and spills while keeping them organized. In addition, consider a second storage box for your original software disks, which you can store in a room away from your computer.

Trap: *You can damage your disks by exposing them to excessive heat or direct sunlight.*

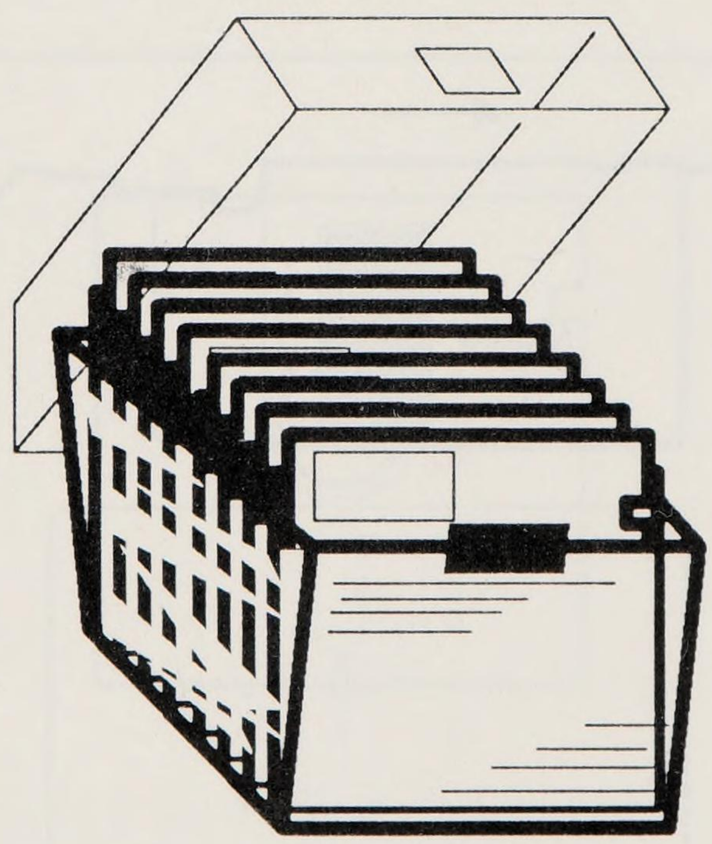
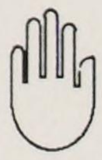


Figure 1-9. Disks in a disk storage box

Although your floppy disks are meant to exist in a variety of environments, they do not like temperatures that are too hot or too cold. The reason for this is simple. If the temperature changes too much, your disks can expand or contract, losing their magnetized contents. Therefore, do not exceed the temperature extremes of 50°F to 110°F.



Trap: *Writing on labels adhered to your disks can cause damage.*

Placing labels on your disks is a very important organizational step. However, never write on a label that is already adhered to your disk. In so doing, you greatly risk damaging the disk and losing the information that it contains. Instead, write on a new label and attach it over the old one.

Trap: *Bending your floppy disks will damage them, destroying the information that they contain.*

Although it seems obvious not to bend your disks while they are outside of your computer, many users inadvertently bend their disks when they insert or remove them from the drives. Remember that your disks contain your information, so treat them with care.

To get started with DOS, you will need to place the DOS Program Disk into drive A. Depending upon your disk drive type, you will have to open this disk drive and remove any disk that the drive may contain. To learn how to open and close your disk drive on your computer, examine the diagrams shown in Figure 1-10.

Tip: *To insert a disk into a drive, ensure that the label is facing up and on the part of the disk nearest you.*

If your disk contains a label, insert the disk into the disk drive with the label facing up and nearest you as shown in Figure 1-11.

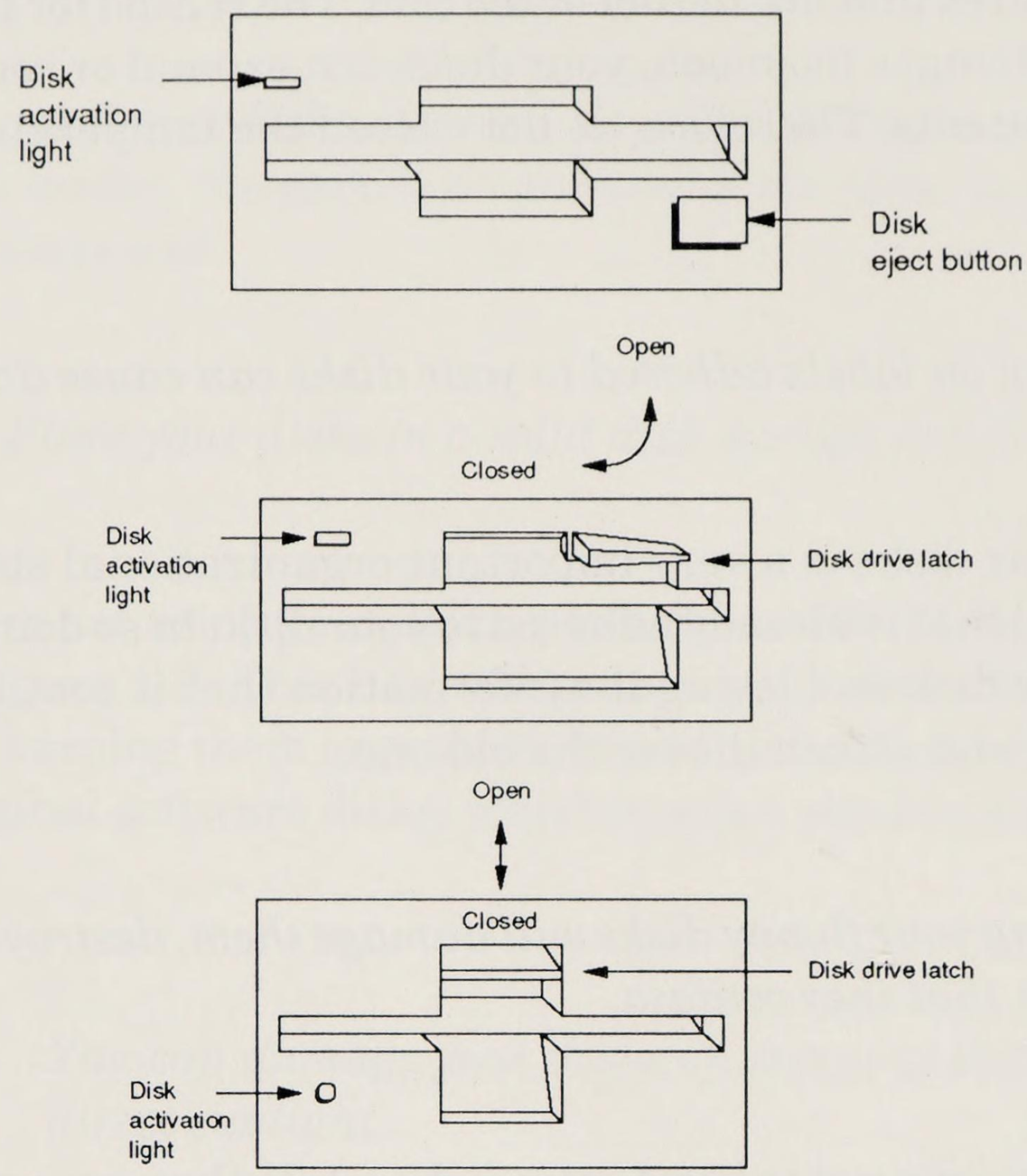


Figure 1-10. Types of disk drives

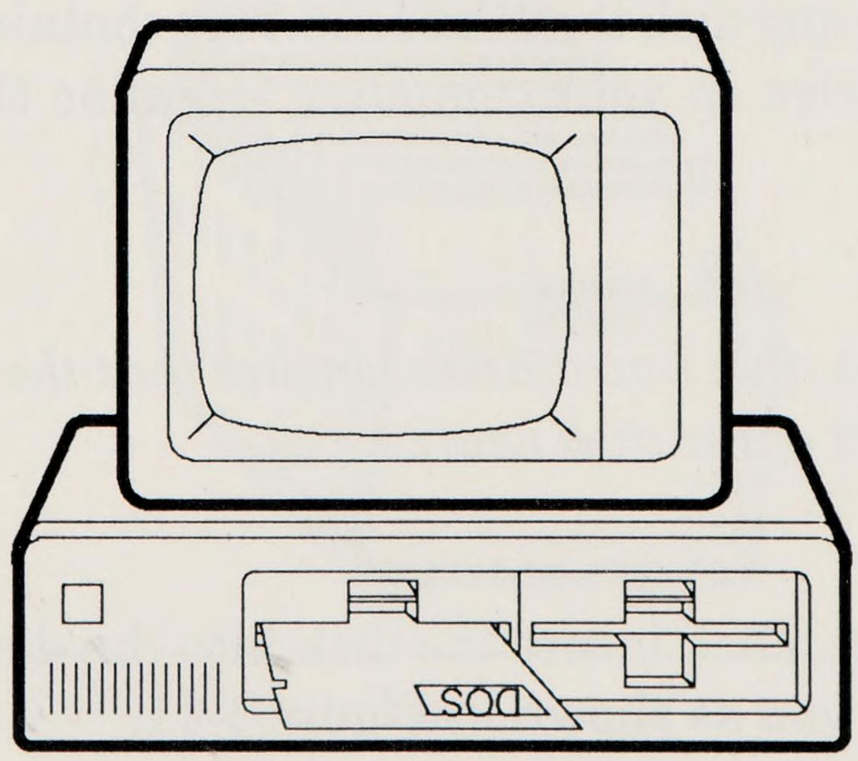
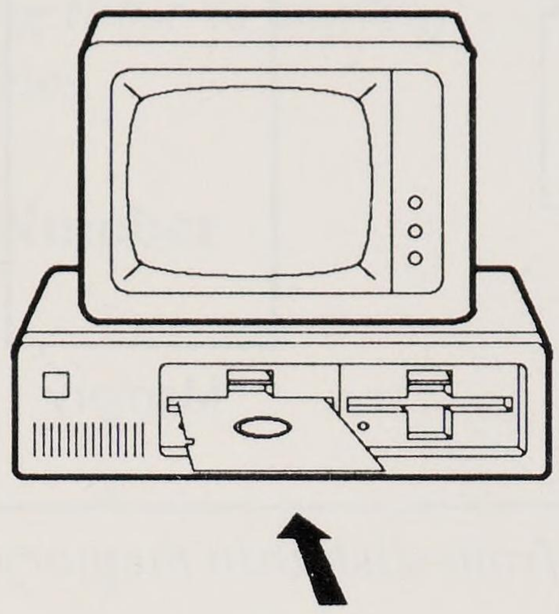


Figure 1-11. Inserting disk into drive

If you are using a microfloppy disk, also insert the disk so that the label is facing up and inserted last. If your disk is unlabeled, insert it with the write-protect notch on the top-left near side as shown here.



If you are using unlabeled microfloppy disks, place the side of the disk containing the *disk spindle* (see Figure 1-12) down with the end of the disk containing the disk shutter inserted first.

Once you have the DOS Program disk inserted in drive A, you are ready to start DOS. To do so, simply turn your computer's power off, wait about ten seconds, and turn the power back on. Your system will again perform its power on self test as discussed previously. If successful, you will see the disk drive ac-

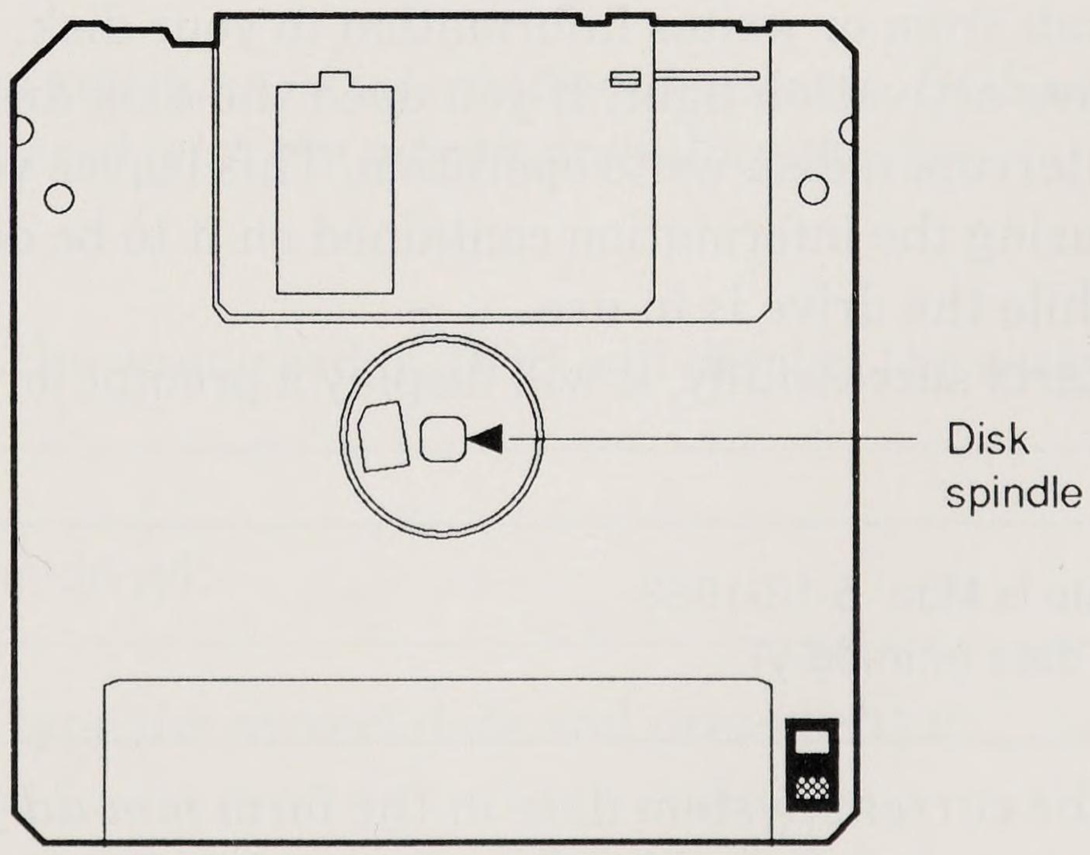


Figure 1-12. A microfloppy disk

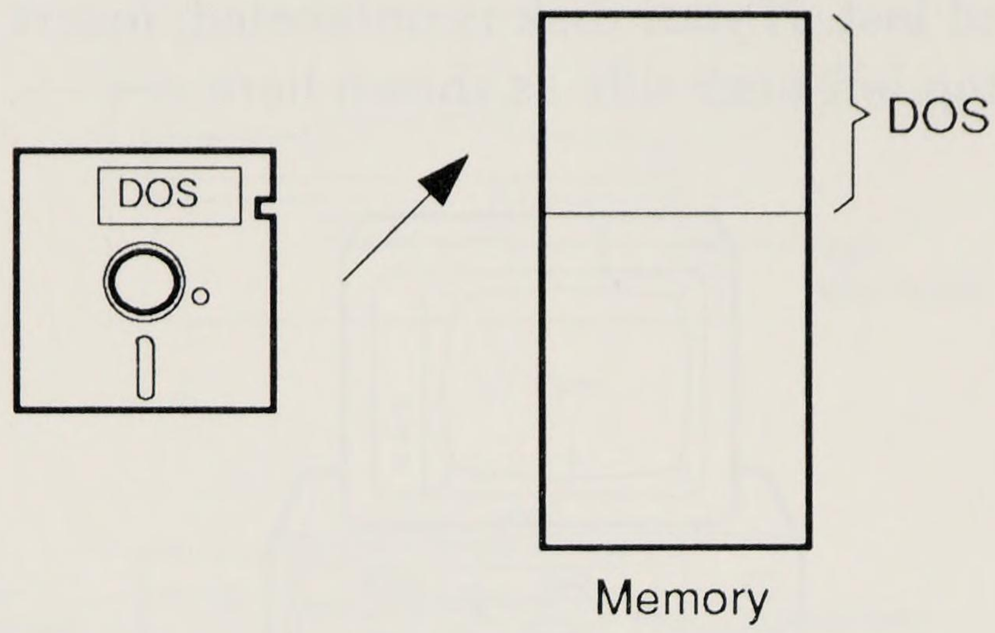


Figure 1-13. Reading DOS from disk into memory

tivation light for drive A illuminate while your computer reads DOS from disk into memory (see Figure 1-13).

Trap: *Opening a disk drive while the activation light is lit may destroy the contents of your disk.*

When DOS reads from or writes information to your disk, your computer turns on the disk drive activation light. If you open the disk drive while this light is on, you may interrupt a disk write operation. This leaves your disk in an incomplete state, causing the information contained on it to be destroyed. Never open a disk drive while the drive is in use.

If DOS starts successfully, it will display a prompt for you to enter the current date.

```
Current date is Mon  6-13-1988
Enter new date (mm-dd-y):
```

Type in the current system date in the form *mm-dd-yy* where *mm* stands for the current month (1-12), *dd* is the current day (1-31) and *yy* is the current year. Assuming, for example, that the date is December 8, 1988, your entry becomes

```
Current date is Mon 6-13-1988
Enter new date (mm-dd-yy): 12/08/88
```

Tip: *Use the following table to associate months with their corresponding numeric values.*

Month	Number
January	1
February	2
March	3
April	4
May	5
June	6
July	7
August	8
September	9
October	10
November	11
December	12

Trap: *With the exception of international systems, DOS expects the date to be entered with the month preceding the day.*

If you enter a date in the wrong order, DOS will display the error message

```
Invalid date
Enter new date (mm-dd-yy):
```

If this occurs, simply type the correct date and press ENTER.

Tip: *If DOS displays the correct date at system startup, simply press the ENTER key to leave the date unchanged.*

Once you have successfully entered the current system date, DOS will prompt you for the current time. DOS expects its time in military 24-hour clock format. Type in the time in the format *hh:mm:ss.nnn* where *hh* is the current hour, *mm* is the current minute, *ss* is the current second, and *nnn* are hundredths of seconds.

Tip: *Use the information in Table 1-1 to convert standard time to military time.*

If you enter an invalid time, DOS will display the error message

```
Invalid time
Enter new time:
```

Simply type the correct time and press the ENTER key.

Tip: *DOS requires you to enter only the hours and minutes at the current time prompt.*

Although DOS allows you to enter the time accurate to hundredths of seconds, it requires only that you enter the hours and minutes, as shown here:

```
Current time is 17:18:35.00
Enter new time: 14:30
```

Tip: *If the time that DOS displays at system startup is correct, simply press ENTER to leave the current time unchanged.*

If you are using an IBM PC/AT, your computer has a built-in clock that remembers the current date and time. If you are using DOS version 3.3, the DATE and TIME commands set your built-in time clock. If you are not using DOS version

Table 1-1. Standard Time Versus Military Time

Standard Time	Military Time
12 Noon	12
1 PM	13
2 PM	14
3 PM	15
4 PM	16
5 PM	17
6 PM	18
7 PM	19
8 PM	20
9 PM	21
10 PM	22
11 PM	23
12 Midnight	00
1 AM	1
2 AM	2
3 AM	3
4 AM	4
5 AM	5
6 AM	6
7 AM	7
8 AM	8
9 AM	9
10 AM	10
11 AM	11

3.3, you will need to use the SETUP disk that accompanied your AT to set the current system date and time.

Tip: *Always make sure that the system date and time are correct.*

Each time that you create or modify a file, DOS marks the file with the current date and time. Thus, you can determine when a file was created or updated by using the DOS DIR command. If your system has the wrong date and time, your file date and time stamps will be invalid.

Trap: *If, rather than displaying the prompt for the DOS date and time, DOS displays the following message, it is telling you that you very likely don't have your DOS disk installed in drive A.*

```
Non-System disk or disk error
Replace and strike any key when ready
```

Before a disk can start DOS, it must contain several operating system files. If your computer displays this message, make sure that the DOS Program Disk is contained in drive A as opposed to the Supplemental Programs Disk. If DOS still won't boot, your DOS Program disk may be damaged.

Once you successfully enter the current date and time, DOS will display its copyright information. If you are using MS-DOS, this message will be in the form

```
Microsoft(R) MS-DOS(R)  Version 3.30
          (C) Copyright Microsoft Corp 1981-1987
```

If you are using PC-DOS, the message will be

```
The IBM Personal Computer DOS
Version 3.30 (C) Copyright International Business Machines Corp
                1981, 1987
                 (C) Copyright Microsoft Corp 1981, 1986
```

Tip: *PC-DOS and MS-DOS are functionally equivalent. The difference lies in the developers.*

PC-DOS is licensed by IBM, while MS-DOS is licensed by Microsoft. Most IBM PCs run PC-DOS; most PC compatibles use MS-DOS. Both versions of DOS support the same series of commands and will appear identical to the user. All of the concepts in this book apply to PC-DOS as well as MS-DOS.

Once DOS displays this copyright information, it will display A>, which is the *DOS prompt.*

```
The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp
1981, 1987
(C)Copyright Microsoft Corp 1981, 1986

A>
```

DOS allows you to run programs by typing in the program's name at the DOS prompt. For example, to clear your screen display, simply type **CLS** and press ENTER.

```
The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp
                1981, 1987
(C)Copyright Microsoft Corp 1981, 1986

A> CLS
```

DOS will clear your screen, displaying the DOS prompt in the upper-left corner of the screen (the *home position*). When you type DOS commands at the DOS prompt in this manner, you can use uppercase or lowercase letters.

```
A>
```

Tip: *Use the DOS DISKCOPY command to make working copies of your DOS disks.*

As discussed earlier in this chapter, you should make copies of all of your original disks. Your DOS disks are no exception. Before you do so, however, you should first *write protect* your original disks.

Tip: *Always write protect your original disks before copying their contents with DISKCOPY.*

If you are using 5 1/4-inch disks, you will find the *write-protect notch* in the upper-right corner of the disk (see Figure 1-14). When this notch is visible, DOS can copy information to your disk and delete old information. In some cases,

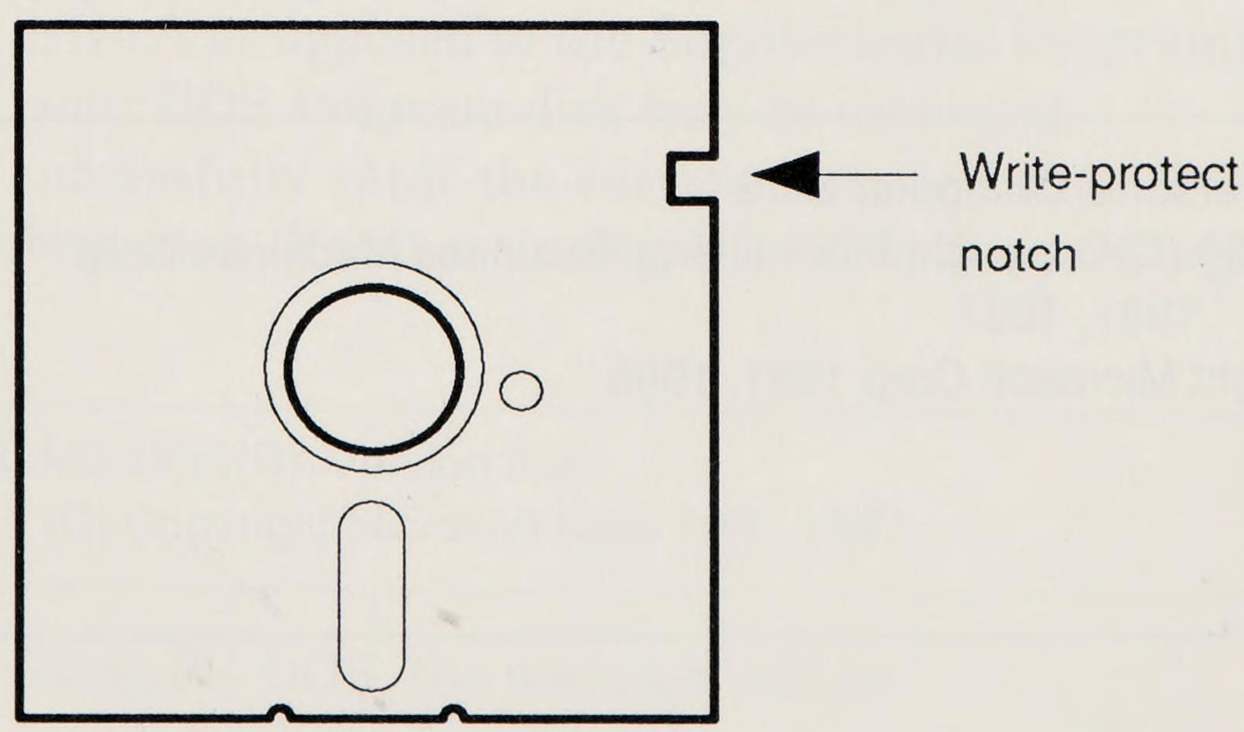


Figure 1-14. Write-protect notch on floppy disk

however, you don't want DOS to update a disk. Simply use one of the write-protect tabs (see Figure 1-15) that are provided with each box of disks that you purchase. This will protect the disk from being written to.

If you are using microfloppy disks, you will find the write- protect opening at the bottom of the disk, as shown in Figure 1-16. When this opening is closed, DOS can upgrade the disk. When it is open, exposing a hole, the disk cannot be modified. Write protect your original DOS disk at this time.

Next, if you have two floppy disk drives, place your DOS Program Disk in drive A and an unused disk in drive B. Issue the following DISKCOPY command.

```
A> DISKCOPY A: B:
```

Trap: *If DOS displays the following message it could not find a command matching the name that you typed.*

```
Bad command or file name
```

First, double-check your typing. If the command that you entered is correct, make sure that you have the disk that you used to start DOS in drive A. If so, DOS should begin copying the contents of your original disk to the disk in drive B as shown in Figure 1-17.

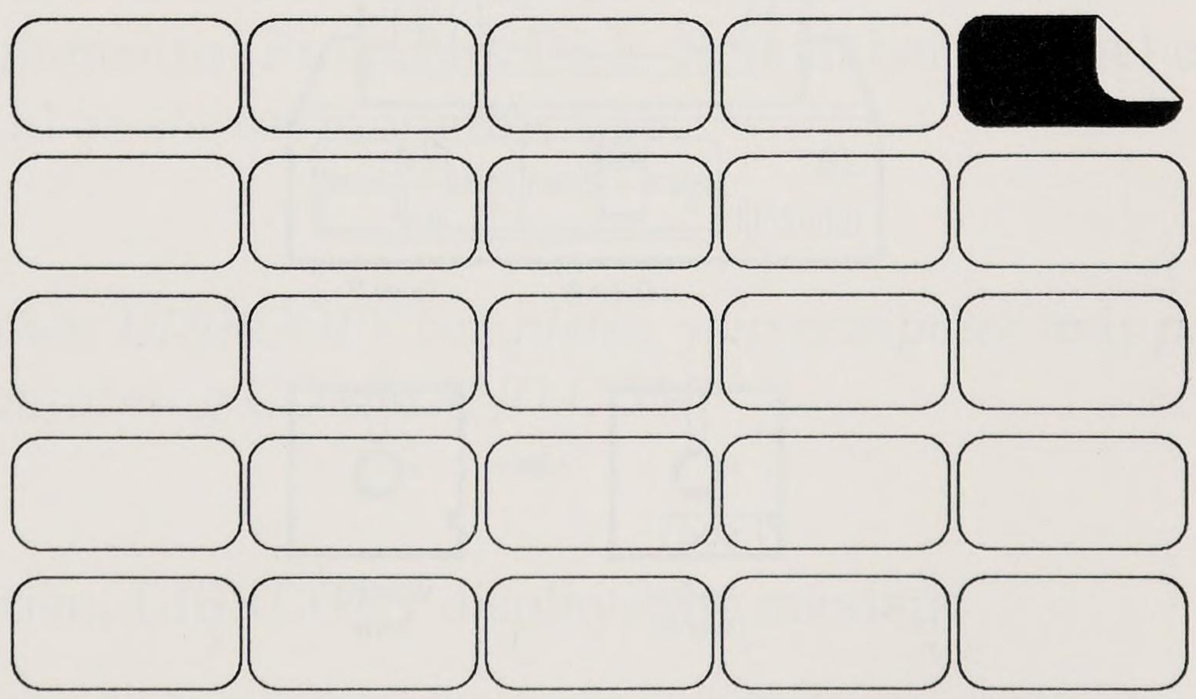


Figure 1-15. Write-protect tabs

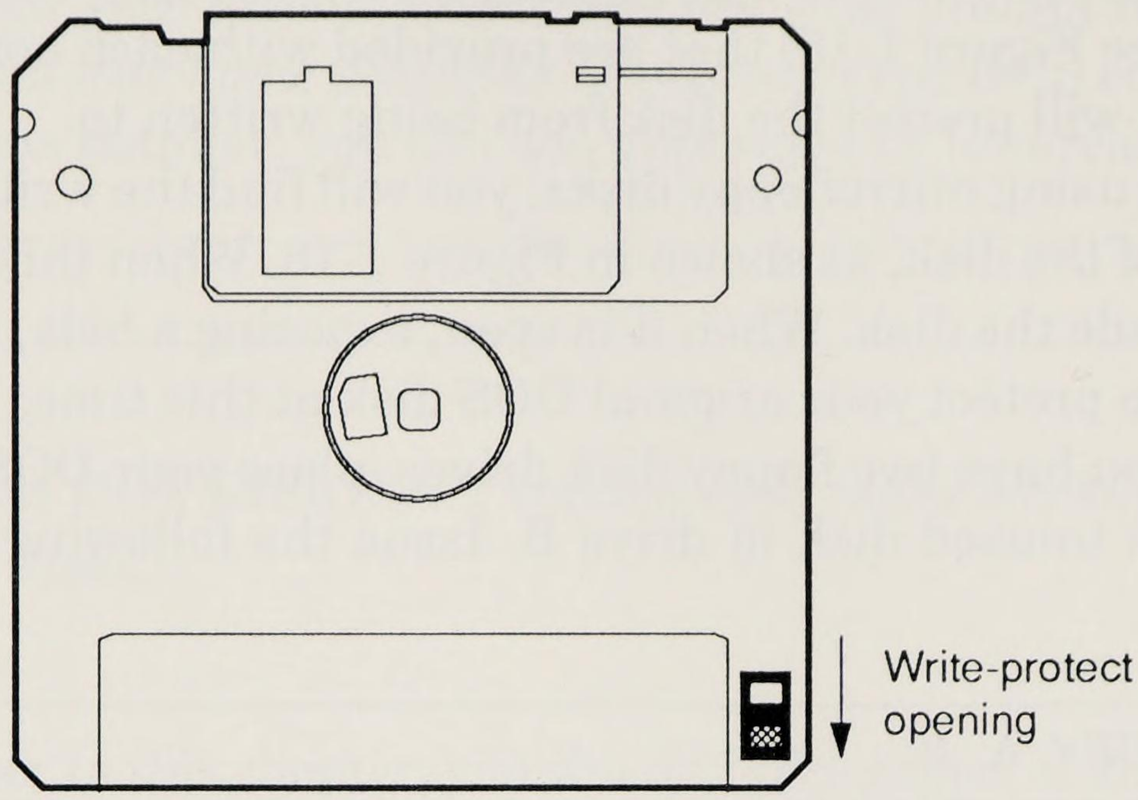


Figure 1-16. Write-protect opening

Trap: *If the new disk in drive B is damaged, DISKCOPY will not complete.*

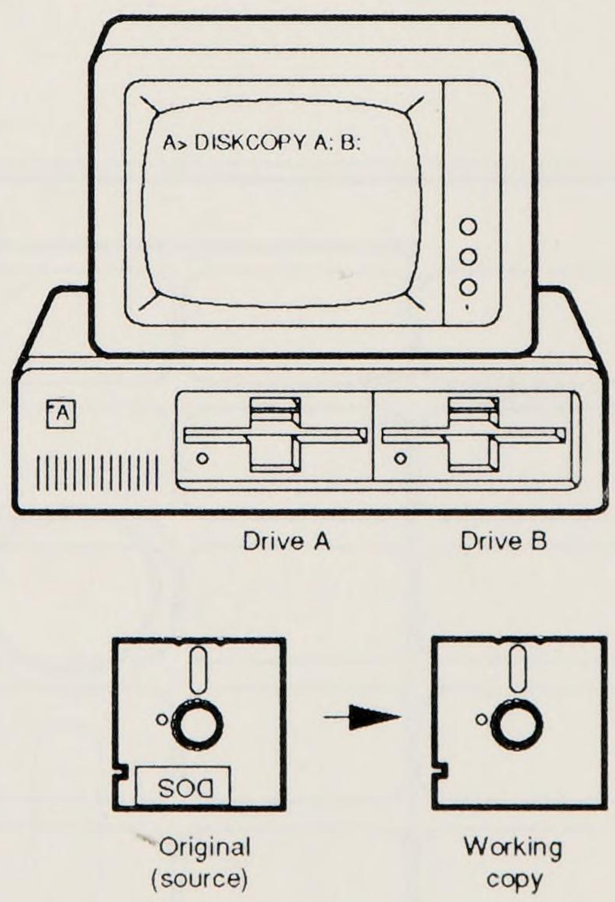


Figure 1-17. DOS copying contents of disk in drive A to disk in drive B

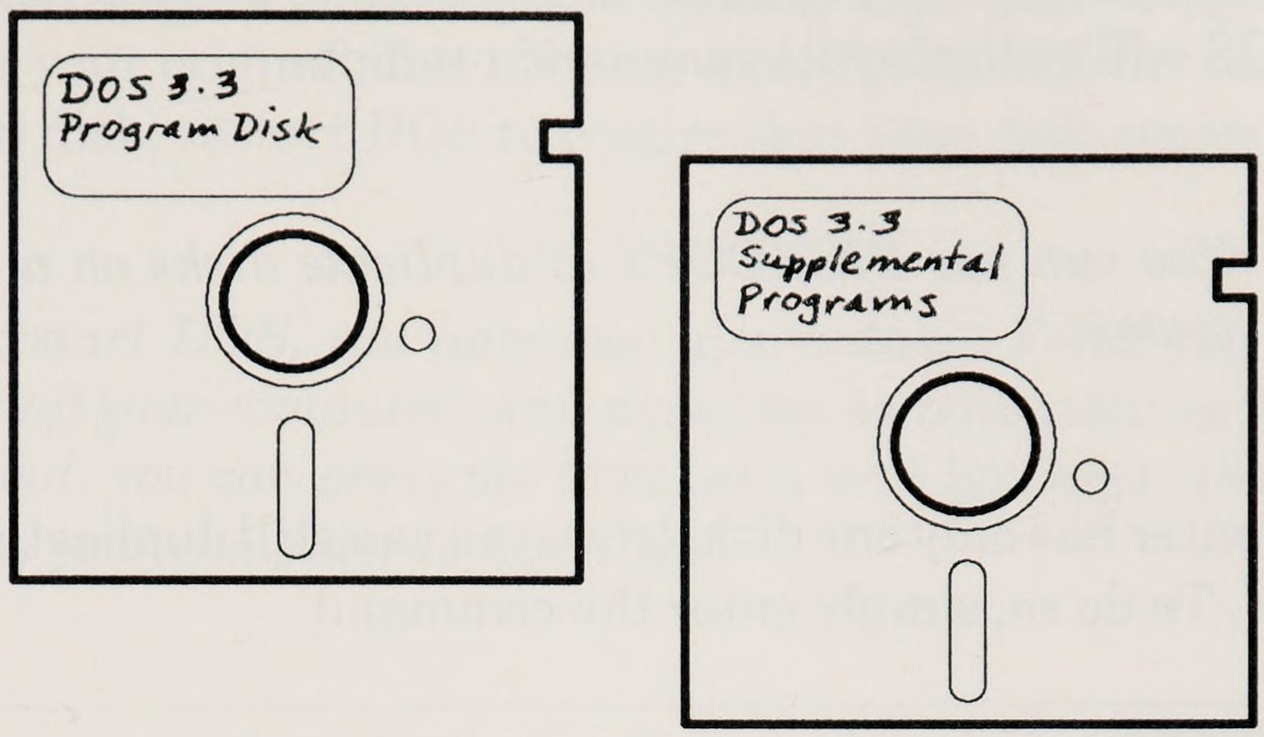


Figure 1-18. Disk labels

Although your disk is new, it may contain damaged locations that cannot be used. If this is the case, DISKCOPY is unable to make an identical copy of your disk and will display the message

```
TARGET disk bad or incompatible
Copy process ended
```

If DISKCOPY displays this message, simply reissue the DISKCOPY command with a new disk in drive B. Repeat the DISKCOPY operation to duplicate your DOS Supplemental Programs Disk. Remember to label each disk (don't write on the disk) as shown in Figure 1-18.

Trap: *When DISKCOPY completes, your computer may prompt for a disk containing COMMAND.COM.*

If, upon completion, DISKCOPY displays the message

```
Insert disk with \COMMAND.COM in drive A
and strike any key when ready
```

simply place your original DOS disk in drive A and press ENTER. Once you have done so, DOS will redisplay its command prompt.

Tip: *You can use DISKCOPY to duplicate disks on a single-disk drive system.*

If your computer has only one disk drive, you can still duplicate your disks using DISKCOPY. To do so, simply enter the command

```
A> DISKCOPY
```

DISKCOPY will respond with

```
Insert SOURCE disk in drive A:

Press any key when ready . . .
```

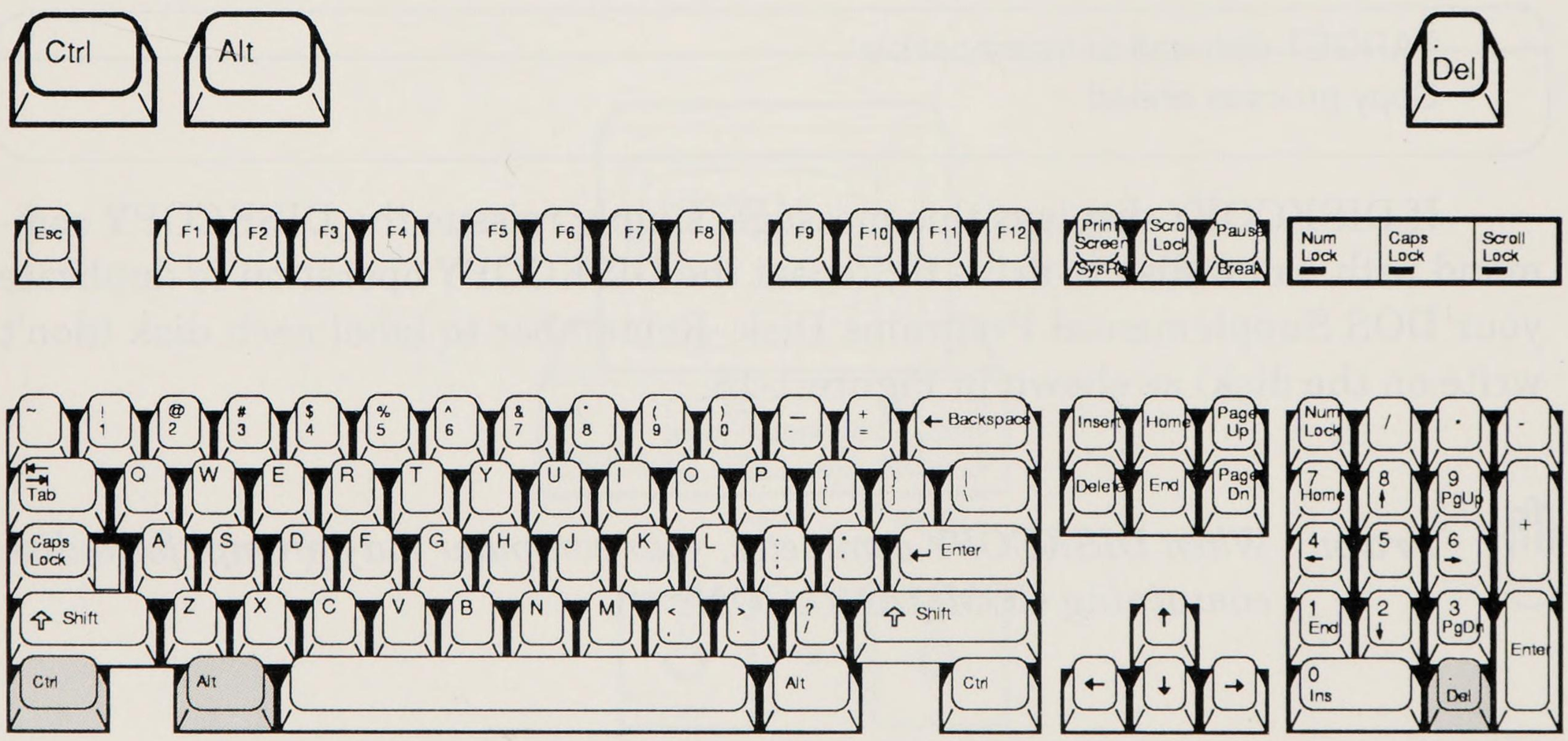


Figure 1-19. *CTRL-ALT-DEL key combination*

Your *source disk* is the disk that you want to copy. Your target disk is the unused disk that your original disk is being copied to. Using the newly created copy of your DOS disk, restart DOS to ensure that your disk copies succeeded.

Tip: *To restart DOS, you have two alternatives. First, you can simply turn off your computer, wait about ten seconds, and turn it back on. Second, you can press the* CTRL, ALT, *and* DEL *keys simultaneously while your computer is running.*

To restart DOS using your keyboard, simultaneously hold down the keys labeled CTRL, ALT, and DEL, as shown in Figure 1-19.

DOS should now restart and again prompt you to enter the current date and time.

```
Current date is Mon  6-13-1988
Enter new date (mm-dd-yy):
Current time is 17:18:35.00
Enter new time: 14:66:30

The IBM Personal Computer DOS
Version 3.30 (C) Copyright International Business Machines Corp 1981, 1987
                (C) Copyright Microsoft Corp 1981, 1986


A>
```

Later in this book you will experiment with DOS commands that require you to restart DOS. Keep in mind that you can now simply restart DOS with the CTRL-ALT-DEL key combination.

Now you are ready to examine the DOS commands found in Chapter 2.

Tips and Traps Summary

Tip: Use the previous illustrations to locate and identify the disk drives present on your system.

Trap: Using multiple-plug adapters for your computer equipment may create a fire hazard and make your computer susceptible to power surges.

Starting Your System

Tip: Place a surge suppressor between your computer equipment and your wall outlet.

Tip: If your computer is not working correctly, ask yourself the following questions.

Does the wall outlet work?
Is the monitor (screen display) plugged in, turned on, and working?
Can you hear the computer's fan once the computer is powered on?
Does your computer display the amount of memory it locates?
Does the disk drive activation light illuminate for drive A?
Is the keyboard properly connected?
Are any error messages displayed?

Tip: Starting DOS is often called booting DOS. Don't be confused by these terms. Both simply refer to the process of loading DOS into your computer's memory from disk.

Tip: If your version of DOS shipped with two disks, you will start (or boot) DOS from the disk labeled "Program Disk."

Tip: Always keep your disks in their disk envelopes when you are not using them.

Tips and Traps Summary (*continued*)

Trap: Touching your disk media can destroy your disk's contents.

Trap: Magnetic and powerful electronic devices can destroy the information on your disks.

Trap: Your phone can destroy the information stored on your disks.

Tip: Always make a duplicate copy of each of your disks.

Trap: Smoking near your computer or disks can destroy your disks.

Tip: Place your disks in a solid disk storage box for safety.

Trap: You can damage your disks by exposing them to excessive heat or direct sunlight.

Trap: Writing on labels adhered to your disks can cause damage.

Trap: Bending your floppy disks will damage them, destroying the information that they contain.

Tip: To insert a disk into a drive, ensure that the label is facing up and on the part of the disk nearest you.

Trap: Opening a disk drive while the activation light is lit may destroy the contents of your disk.

Tip: Associate months with their corresponding numeric values.

Trap: With the exception of international systems, DOS expects the date to be entered with the month preceding the day.

Tip: If DOS displays the correct date at system startup, simply press the ENTER key to leave the date unchanged.

Tip: Use the information in Table 1-1 to convert standard time to military time.

Tips and Traps Summary (*continued*)

Tip: DOS requires you to enter only the hours and minutes at the current time prompt.

Tip: If the time that DOS displays at system startup is correct, simply press ENTER to leave the current time unchanged.

Tip: Always make sure that the system date and time are correct.

Trap: If, rather than displaying the prompt for the DOS date and time, DOS displays the following message, it is telling you that you very likely don't have your DOS disk installed in drive A.

```
Bad command or filename
```

Tip: PC-DOS and MS-DOS are functionally equivalent. The difference lies in the developers.

Tip: Use the DOS DISKCOPY command to make working copies of your DOS disks.

Tip: Always write protect your original disks before copying their contents with DISKCOPY.

Trap: If DOS displays the following message, it could not find a command matching the name that you typed.

```
Bad command or filename
```

Trap: If the new disk in drive B is damaged, DISKCOPY will not complete.

Tips and Traps Summary (*continued*)

Trap: When DISKCOPY completes, your computer may prompt for a disk containing COMMAND.COM.

Tip: You can use DISKCOPY to duplicate disks on a single-disk drive system.

Tip: To restart DOS, you have two alternatives. First, you can simply turn off your computer, wait about ten seconds, and turn it back on. Second, you can press the CTRL, ALT, and DEL keys simultaneously while your computer is running.

2

Entering DOS Commands

In Chapter 1 you learned how to start DOS and make working copies of your DOS disks. As with any new software package, you should place your original disks in a safe location. From this point on, you should be using your working copies of DOS on a daily basis, not your original disks.

If your computer is off, start DOS as explained in Chapter 1. DOS will again prompt you to enter the current date and time. Next, it will display its copyright information and then the DOS prompt, as shown here:

```
Current date is Mon  6-13-1988
Enter new date (mm-dd-yy):
Current time is 17:18:35.00
Enter new time: 14:66:30

The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp
                1981, 1987
                (C)Copyright Microsoft Corp 1981, 1986

A>
```

As discussed in Chapter 1, A> is the DOS prompt. This prompt is simply DOS's way of telling you that it is ready for you to enter a command. If you press the ENTER key, DOS will respond by displaying a second prompt, as shown here:

```
Current date is Mon  6-13-1988
Enter new date (mm-dd-yy):
Current time is 17:18:35.00
Enter new time: 14:66:30

The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp
                1981, 1987
                (C)Copyright Microsoft Corp 1981, 1986


A>
A>
```

Pressing the ENTER key several times results in

```
Current date is Mon  6-13-1988
Enter new date (mm-dd-yy):
Current time is 17:18:35.00
Enter new time: 14:66:30

The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp
                1981, 1987
                (C)Copyright Microsoft Corp 1981, 1986


A>
A>
A>
A>
A>
A>
```

In Chapter 1 you used the DOS CLS command to clear the screen and the DOS DISKCOPY command to duplicate disks. To do so, you simply typed in the command name and pressed the ENTER key. Remember that although there are two common versions of DOS, PC-DOS and MS-DOS, they are functionally identical. As you have seen, each time you start up your system, DOS displays its copyright information on the screen, which reveals whether you are using PC-DOS or MS-DOS.

```
The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp
                1981, 1987
                (C)Copyright Microsoft Corp 1981, 1986
```

A second method of determining which version of DOS you are using is to invoke the DOS VER (version) command.

Tip: *Although PC-DOS and MS-DOS are functionally identical, you may encounter differences between early versions. To determine which DOS version you are executing, use the DOS VER command.*

To invoke VER, type **VER** at the DOS prompt:

```
A> VER
```

For PC-DOS version 3.3, DOS will display

```
IBM Personal Computer DOS Version 3.30
```

Likewise, for MS-DOS version 3.3, your output will be

```
MS-DOS Version 3.30
```

Tip: *Each DOS version number is comprised of two parts, a major version number and a minor version number.*

If you have DOS version 3.2, for example, the major version number is 3 while the minor version number is 2. When the DOS developers upgrade DOS, they increment the version number as follows.

- If the upgrade is a minor change to the operating system for bug fixes and minor enhancements, the developers will increment the minor version number. In this case, DOS version 3.2 would become DOS version 3.3.
- If the upgrade is a major change to the operating system, such as increasing the functionality and introducing new commands, the developers will increment the major version number. In this case, DOS version 3.3 would become DOS version 4.0.

Most DOS users upgrade to a new version of DOS only when a major release of the operating system occurs. Table 2-1 summarizes the upgrades to MS-DOS and PC-DOS that have occurred since their release in 1981.

Although most of the commands in this book execute under all versions of DOS, an occasional command qualifier will be version specific. Remember that the DOS VER command allows you to determine which version of DOS you are executing.

Internal Versus External DOS Commands

Tip: *DOS commands are categorized as one of two types, either internal or external. Internal commands always reside in your computer's memory, while external commands reside on disk.*

When you enter commands at the DOS prompt, you must keep in mind whether the command is external or internal. An external command resides on disk; an internal command is always present in memory. By understanding the com-

Table 2-1. Versions of DOS

Version	Date	Functionality
	MS-DOS	
1.0	1981	Original disk operating system
1.25	1982	Support for double-sided disks
2.0	1983	Support for subdirectories
2.01	1983	Support for international symbols
2.11	1983	Bug fixes
2.25	1983	Extended character set support
3.0	1984	Support for 1.2MB floppy disk
		Support for larger hard disk
3.1	1984	Support for PC networks
3.2	1986	Support for microfloppies
3.3	1987	Support for PS/2 series computers
	PC-DOS	
1.0	1981	Original disk operating system
2.0	1983	Support for subdirectories
2.10	1983	Bug fixes
3.0	1984	Support for 1.2MB floppy disk
		Support for larger hard disk
3.1	1984	Support for PC networks
3.2	1986	Support for microfloppies
3.3	1987	Support for PS/2 series computers

mand types (internal and external), you will understand why your DOS disk must be in drive A for some commands while not for others.

Each time you start DOS, it loads several commands into memory that remain there during your current session. Since these commands are always

present in memory, they are called *internal commands.* You have already issued two internal DOS commands—CLS and VER (see Figure 2-1).

The following table lists the internal DOS commands:

BREAK	DIR	RMDIR
CHDIR	ERASE	SET
CLS	EXIT	TIME
COPY	MKDIR	TYPE
CTTY	PATH	VER
DATE	PROMPT	VERIFY
DEL	REN	VOL

The internal DOS commands comprise only a portion of the DOS commands. A program must reside in memory before the computer can execute it. The DOS developers had to restrict the number of commands that always reside in memory (internal commands) because the programs consume too much memory. Therefore, DOS places a portion of its common commands in memory, while keeping the larger programs on disk.

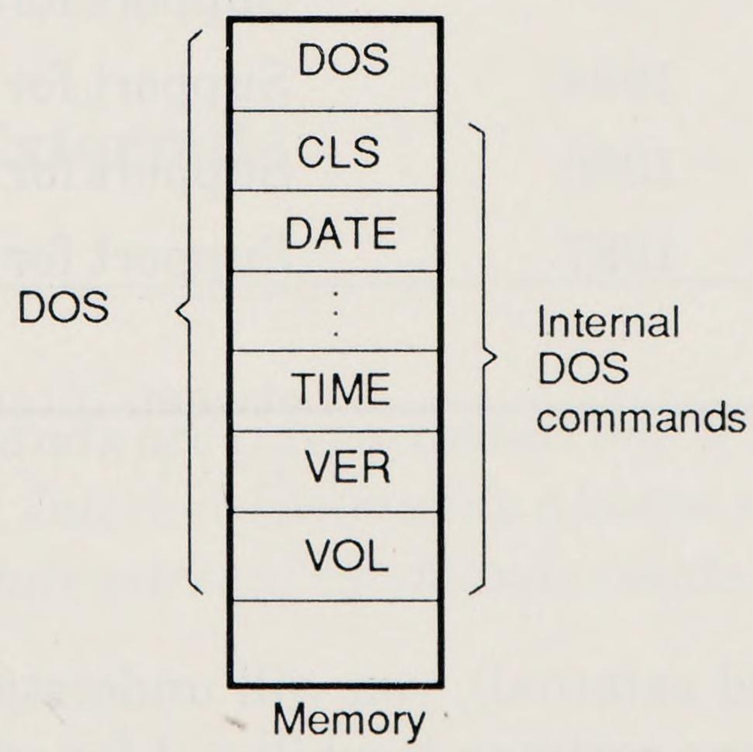


Figure 2-1. Internal DOS commands in memory

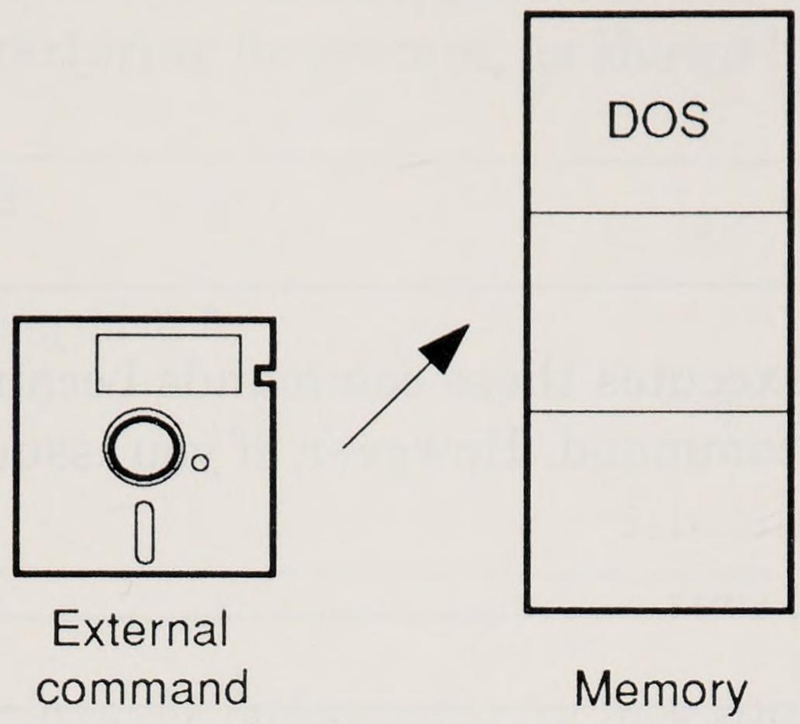


Figure 2-2. Execution of external DOS command

Commands that reside on disk as opposed to in memory are called *external DOS commands.* When you execute an external command, DOS loads the program from disk into memory in order to execute the command (see Figure 2-2). Before you can execute an external DOS command, the disk containing the command must be in your disk drive.

Trap: *If DOS cannot find an external command on disk, it will display the following error message:*

```
Bad command or file name
```

To better understand the difference between internal and external commands, open your disk drive and remove your DOS disk. Next, issue the DOS CLS command

```
A> CLS
```

DOS will clear your screen display, placing the DOS prompt in the upper-left corner of the screen. In a similar manner, issue the DOS VER command

```
A> VER

MS-DOS Version 3.30

A>
```

DOS successfully executes these commands because they are present in memory as an internal command. However, if you issue the DOS DISKCOPY command

```
A> DISKCOPY  A:  B:
```

DOS will display

```
A> DISKCOPY  A:  B:

Not ready error reading drive A
Abort, Retry, Fail?
```

Trap: *When DOS cannot access a disk in the specified drive, it will display the following message:*

```
Not ready error reading drive A
Abort, Retry, Fail?
```

Because DISKCOPY is not an internal DOS command, DOS searched for it on disk. However, since the disk drive door is open and the DOS disk is not present, DOS cannot access the disk. Since DOS cannot continue without your intervention (inserting a disk and closing the disk drive latch), it is asking you how it should continue.

Tip: *If DOS displays the following message and you want to terminate the command causing the error, type* **A** *(for Abort) and press* ENTER*:*

```
Not ready error reading drive A
Abort, Retry, Fail?
```

The letter "A" directs DOS to abort the current command. DOS will terminate the command, restoring its prompt, as shown here:

```
A> DISKCOPY A: B:

Not ready error reading drive A
Abort, Retry, Fail? A

A>
```

Tip: *If DOS displays the following error message and you want it to retry the operation that caused the error after you have intervened (by inserting a disk in the drive), type* **R** *and press* ENTER:

```
A> DISKCOPY A: B:

Not ready error reading drive A
Abort, Retry, Fail?
```

The letter "R" in this case stands for Retry. Unless you have taken steps to correct the error in some manner, DOS will simply display the error message once again, as shown here:

```
A> DISKCOPY A: B:

Not ready error reading drive A
Abort, Retry, Fail? R

Not ready error reading drive A
Abort, Retry, Fail?
```

However if you insert a disk in drive A and close the disk drive latch, pressing **R** results in

```
A> DISKCOPY A: B:

Not ready error reading drive A
Abort, Retry, Fail? R
```

```
Insert SOURCE diskette in drive A:

Press any key when ready . . .
```

Tip: *If DOS displays the following message and you have a disk in the specified drive with the disk drive latch closed, type* **F** *and press ENTER (assuming the Retry option has already failed):*

```
A> DISKCOPY A: B:

Not ready error reading drive A
Abort, Retry, Fail?
```

In rare instances, DOS may encounter errors when accessing your disk. If you are lucky, you may be able to copy some of the files on the disk, saving them to a new disk by using the F option.

In older DOS versions, this error message may appear with an I (Ignore error) option in place of F, as shown here:

```
Abort, Retry, Ignore?
```

If your version of DOS displays this message, simply respond to it in the same manner as just discussed, replacing the Ignore option for Fail.

The Default Disk Drive

Tip: *Regardless of the number of disk drives present on your system, DOS selects one disk drive as your current, or default, drive.*

As previously discussed, DOS external commands reside on disk. Each time that you invoke an external DOS command, DOS loads the command from disk into memory. When you previously issued the DOS DISKCOPY command, DOS looked for the command on drive A. DOS searched only drive A for DISKCOPY,

despite the fact that your computer may have a second floppy-disk drive (drive B) or a fixed disk (drive C). DOS searched drive A for DISKCOPY because drive A is the current, or default, disk drive.

Tip: *DOS defines the default disk drive as the drive that it will search, by default, for external commands.*

Don't let the definition of default drive confuse you. The default drive exists to save you keystrokes when you enter your DOS commands. Unless you tell DOS otherwise (by specifying a disk drive in the form B:DISKCOPY), DOS will search the current drive for external commands by default. If your program resides on a disk in a drive other than the default drive, you simply specify the disk drive to DOS, as shown here:

```
A> B:DISKCOPY A: B:
```

or

```
A> C:FORMAT A:
```

Tip: *The DOS prompt contains your current (default) disk drive.*

Unless you change the DOS prompt (with the DOS PROMPT command discussed later in this book), it will contain your default disk drive as shown here:

A>

Default disk drive

Tip: *To change your default drive, simply type in the letter of the drive desired followed immediately by a colon (B:).*

You will want to select a new disk drive as the default when the drive contains a disk with several data files or programs that you need. To do so, simply type the letter associated with the disk drive followed immediately by a colon, as shown here:

```
A> B:
```

When you press the ENTER key, DOS will acknowledge your change in default drives by updating your DOS prompt, as shown here:

```
A> B:
B>
```

DOS will then search drive B, by default, for files and external DOS commands.

Setting the System Date and Time

In Chapter 1 you learned how to set your computer's date and time when you start DOS, along with the importance of doing so. If after DOS starts you realize that your current system date or time is incorrect, you can reset them with the DOS DATE and TIME commands.

Tip: *If your system date is incorrect, or if you simply want to display the current date, issue the DOS DATE command.*

DATE is an internal DOS command that allows you to set your computer's current date. To invoke DATE, type **DATE** at the DOS prompt, as shown here:

```
A> DATE
```

When you press ENTER, DOS will acknowledge your command by prompting you to enter the current date, as shown here:

```
A> DATE
Current date is Tue  6-14-1988
Enter new date (mm-dd-yy):
```

As before, simply type the current date in the form *mm-dd-yy*. If you want to leave the current date unchanged, press ENTER, returning to the DOS prompt.

```
A> DATE
Current date is Tue  6-14-1988
Enter new date (mm-dd-yy):

A>
```

Tip: *You can determine on which day of the week a particular date falls by using the DOS DATE command.*

Each time you invoke the DOS DATE command, DOS displays the current day of the week followed by the date, as shown here:

```
A> DATE
Current date is Tue  6-14-1988
Enter new date (mm-dd-yy):
```

If, for example, you want to determine on which day of the week Christmas falls in 1989, invoke DATE and change the system date to 12/25/89.

```
A> DATE
Current date is Tue  6-14-1988
Enter new date (mm-dd-yy): 12/25/89

A>
```

Next, again issue the DATE command.

```
A> DATE
Current date is Mon 12-25-1989
Enter new date (mm-dd-yy):
```

DOS displays the day of the week (Mon) when Christmas will fall as desired. Make sure that you set the system date back to the current date before proceeding.

In a manner similar to setting the current date, you can change the current system time by issuing the DOS TIME command.

Tip: *If your system time is incorrect, or if you simply want to display the current time, issue the DOS TIME command.*

TIME is an internal DOS command that allows you to set your computer's current time. To invoke TIME, type **TIME** at the DOS prompt, as shown here:

```
A> TIME
```

When you press ENTER, DOS will acknowledge your command by prompting you to enter the current time, as shown here:

```
A> TIME
Current time is 19:16:07.49
Enter new time:
```

As before, type the current time in the form *hh:mm:ss.nnn*. If you want to leave the current time unchanged, press ENTER, returning to the DOS prompt.

```
A> TIME
Current time is 19:16:07.49
Enter new time:

A>
```

Tip: *Many DOS commands support optional command-line parameters that can simplify command execution.*

Each time that you issue a DOS command, the command that you type becomes your DOS command line. For example, if you enter the command

```
A> VER
```

your command line is the VER command. However, many DOS commands allow you to enter optional values in the command line. For example, the DOS DATE command allows you to specify the current date when you invoke the command, as shown here:

```
A> DATE 12/08/88
```

As long as the date that you specify is a valid date, DOS will update the system date, redisplaying the DOS prompt, as shown here:

```
A> DATE 12/08/88
A>
```

However, if you enter an invalid date, DOS will display an error message and prompt you to enter the correct date, as shown here:

```
Invalid date
Enter new date (mm-dd-yy):
```

To determine which DOS commands support optional command-line parameters, review the command reference section (Appendix B) of this book. Items that appear within brackets [ ] are optional as shown in the DATE command format: DATE [mm-dd-yy].

This command format tells you that you can enter the command as

```
A> DATE
```

or you can specify a date in the command line, as shown here:

```
A> DATE 11/04/88
```

Many of the DOS commands examined in this book support optional command-line arguments.

Tips and Traps Summary

Tip: Although PC-DOS and MS-DOS are functionally identical, you may encounter differences between early versions. To determine which DOS version you are executing, use the DOS VER command.

Tip: Each DOS version number is comprised of two parts, a major version number and a minor version number.

Internal Versus External DOS Commands

Tip: DOS commands are categorized as one of two types, either internal or external. Internal commands always reside in your computer's memory, while external commands reside on disk.

Trap: If DOS cannot find an external command on disk, it will display the following error message:

```
Bad command or filename
```

Trap: When DOS cannot access a disk in the specified drive, it will display the following message:

```
Not ready error reading drive A
Abort, Retry, Fail?
```

Tip: If DOS displays the following message and you want to terminate the command causing the error, type **A** (for Abort) and press ENTER.

```
Not ready error reading drive A
Abort, Retry, Fail?
```

Tips and Traps Summary (*continued*)

Tip: If DOS displays the following message and you have a disk in the drive specified with the drive latch closed, type **F** and press ENTER (assuming that the Retry option has already failed).

```
A DISKCOPY A: B:

Not ready error reading drive A
Abort, Retry, Fail?
```

The Default Disk Drive

Tip: Regardless of the number of disk drives present on your system, DOS selects one disk drive as your current, or default, drive.

Tip: DOS defines the default disk drive as the drive that it will search, by default, for external commands.

Tip: The DOS prompt contains your current (default) disk drive.

Tip: To change your default drive, simply type in the letter of the drive desired followed immediately by a colon (B:).

Setting the System Date and Time

Tip: If your system date is incorrect, or if you simply want to display the current date, issue the DOS DATE command.

Tip: You can determine on which day of the week a particular date falls by using the DOS DATE command.

Tips and Traps Summary (*continued*)

Tip: If your system time is incorrect, or if you simply want to display the current time, issue the DOS TIME command.

Tip: Many DOS commands support optional command-line parameters that can simplify command execution.

3

Handling Your Files

Just as you store documents and information in files in your office, you must store the information that you need from one session with your computer to the next in DOS files. Likewise, just as you can create, modify, and get rid of files in your office, DOS allows you to create, copy, modify, rename, and delete files that reside on your disks.

By the end of this chapter you should be fully conversant with DOS file manipulation, DOS file names, and many of the file-manipulation commands that you will use on a daily basis. In addition, you will learn how to manipulate disk files on other than the default drive and how you can save keystrokes in the process. To get started, consider the files provided on your DOS Program Disk.

Looking at Directories

Files exist on your disk for one purpose: to store programs and data. As you will see, DOS provides several files on your DOS disks. Place your DOS Program Disk in drive A and issue the command

```
A> DIR
```

Tip: *The DOS DIR command allows you to display the names of all the files that reside on your disk.*

DOS keeps a list of the files that reside on your disk. When you create a DOS file, DOS adds the file to this list. Likewise, when you delete a file from your disk, DOS removes the file from the list. This list of DOS files is called a *directory*. The DOS DIR (directory) command allows you to display the files that reside on your disk as shown in Figure 3-1.

Tip: *DIR displays a file's name, extension, size, and creation / modification date and time.*

The DOS DIR command not only displays the name of each file on the disk, but also the file's extension, size (in bytes), and the date and time that the file was created or last modified, as shown here:

```
4201          CPI    17089     7-24-87  12:00a
5202          CPI    459       7-24-87  12:00a
ANSI          SYS    1647      7-24-87  12:00a
```

File name

Extension

Size

Modification/Creation Date and Time

```
Volume in drive A is MS330PP01
Directory of  A:\

4201            CPI     17089   7-24-87  12:00a
5202            CPI     459     7-24-87  12:00a
ANSI            SYS     1647    7-24-87  12:00a
APPEND          EXE     5794    7-24-87  12:00a
ASSIGN          COM     1530    7-24-87  12:00a
ATTRIB          EXE     10656   7-24-87  12:00a
CHKDSK          COM     9819    7-24-87  12:00a
COMMAND         COM     25276   7-24-87  12:00a
COMP            COM     4183    7-24-87  12:00a
COUNTRY         SYS     11254   7-24-87  12:00a
DISKCOMP        COM     5848    7-24-87  12:00a
DISKCOPY        COM     6264    7-24-87  12:00a
DISPLAY         SYS     11259   7-24-87  12:00a
DRIVER          SYS     1165    7-24-87  12:00a
EDLIN           COM     7495    7-24-87  12:00a
EXE2BIN         EXE     3050    7-24-87  12:00a
FASTOPEN        EXE     3888    7-24-87  12:00a
FDISK           COM     48919   7-24-87  12:00a
FIND            EXE     6403    7-24-87  12:00a
FORMAT          COM     11671   7-24-87  12:00a
GRAFTABL        COM     6136    7-24-87  12:00a
GRAPHICS        COM     13943   7-24-87  12:00a
JOIN            EXE     9612    7-24-87  12:00a
KEYB            COM     9041    7-24-87  12:00a
LABEL           COM     2346    7-24-87  12:00a
MODE            COM     15440   7-24-87  12:00a
MORE            COM     282     7-24-87  12:00a
NLSFUNC         EXE     3029    7-24-87  12:00a
PRINT           COM     8995    7-24-87  12:00a
RECOVER         COM     4268    7-24-87  12:00a
SELECT          COM     4132    7-24-87  12:00a
SORT            EXE     1946    7-24-87  12:00a
SUBST           EXE     10552   7-24-87  12:00a
SYS             COM     4725    7-24-87  12:00a
        34 File(s)     5120 bytes free
```

Figure 3-1. Directory of files on disk

Tip: *DOS file names are actually comprised of two parts, a file name and an optional extension.*

As you can see from the directory listing, DOS file names are actually comprised of two parts. The first part is an 8-character name. The second part is an optional 3-character extension. To manipulate DOS files, you must separate the file name from the extension with a period. Combining the name and extension of the previous directory listing yields

```
4201.CPI   APPEND.EXE
5202.CPI   ATTRIB.EXE
ANSI.SYS   ASSIGN.COM
```

Tip: *A file's extension is a good guide to the file's contents.*

As you can see from the directory of your DOS Program Disk, most of your files have the extensions COM, EXE, SYS, or CPI. This 3-character extension helps you group files by function. For example, files with the extension COM or EXE are executable DOS commands (such as DISKCOPY); these will be used throughout this book. Files with the extension SYS (which is short for SYSTEM) are operating system files. Lastly, depending upon your version of DOS, you may or may not have files with the extension CPI. These files contain Code Page Information used by international DOS users to support foreign character sets.

This table summarizes the file types found on your DOS Program Disk.

Extension	File Type
COM	DOS command
EXE	Executable DOS program
SYS	Operating system file
CPI	Code page information file

In addition to the file extensions used by DOS, the following extensions are commonly used for files of other types.

Extension	File Type
BAT	DOS batch file
LTR	File containing a letter
RPT	File containing a report
BAK	Backup file
BAS	File containing a BASIC program
TXT	Text file

Usually, by examining a file's extension, you can quickly determine the file's function as shown in the following table:

File Name	File Type
DISKCOPY.COM	DOS command file
ANSI.SYS	Operating system file
JOIN.EXE	Executable DOS program
GRANDMA.LTR	File containing a letter
TEST.TXT	Text file
SALES.RPT	File containing a report

Tip: *DOS displays each file's size in bytes. A byte is approximately one character.*

Do not be confused by the term *byte*. If a file is 256 bytes in length, it contains approximately 256 characters of information.

Tip: *The date and time displayed in the DOS directory listing are the date and time that the file was either created or last modified.*

As discussed in Chapter 1, it is important that you keep your system date and time accurate. Each time you create or modify a file, DOS uses the current date and time to record the file modification. If your system date and time are in-

valid, the date and time DOS associates with your file are essentially meaningless.

Trap: *If a disk contains many files, its directory listing may scroll quickly on the screen, causing you to miss your desired file.*

In most cases, your screen display has enough room for only 25 lines of information. If your disk contains more files than that, DOS will scroll file names off the top of the screen in order to make room for new files at the bottom. If this becomes a problem, use the /P qualifier with the DIR command.

Tip: *DIR provides the /P qualifier, which suspends scrolling of file names with each screenful of information.*

At the DOS prompt, type the following DIR command:

```
A> DIR /P
```

DOS will display the first screenful of files and then pause, prompting you to press a key to continue (see Figure 3-2).

When you press a key, DIR will display the next screenful of files, repeating this process (assuming a full screen of files exists) as shown in Figure 3-3. By using DIR in this manner, you can control the way in which DIR scrolls files on your screen display.

Tip: *If you are not interested in the file size, date, and time fields, you can use the /W qualifier, which directs DIR to suppress the display of those fields.*

Just as the /P option directs DIR to display directory information a screenful at a time, the /W qualifier directs DIR to display file names and extensions with five file names across the screen as shown in Figure 3-4.

```
Volume in drive A is MS330PP01
Directory of  A:\

4201          CPI     17089   7-24-87  12:00a
5202          CPI     459     7-24-87  12:00a
ANSI          SYS     1647    7-24-87  12:00a
APPEND        EXE     5794    7-24-87  12:00a
ASSIGN        COM     1530    7-24-87  12:00a
ATTRIB        EXE     10656   7-24-87  12:00a
CHKDSK        COM     9819    7-24-87  12:00a
COMMAND       COM     25276   7-24-87  12:00a
COMP          COM     4183    7-24-87  12:00a
COUNTRY       SYS     11254   7-24-87  12:00a
DISKCOMP      COM     5848    7-24-87  12:00a
DISKCOPY      COM     6264    7-24-87  12:00a
DISPLAY       SYS     11259   7-24-87  12:00a
DRIVER        SYS     1165    7-24-87  12:00a
EDLIN         COM     7495    7-24-87  12:00a
EXE2BIN       EXE     3050    7-24-87  12:00a
FASTOPEN      EXE     3888    7-24-87  12:00a
FDISK         COM     48919   7-24-87  12:00a
FIND          EXE     6403    7-24-87  12:00a
FORMAT        COM     11671   7-24-87  12:00a
GRAFTABL      COM     613     7-24-87  12:00a
GRAPHICS      COM     13943   7-24-87  12:00a
JOIN          EXE     9612    7-24-87  12:00a
Strike a key when ready . . .
```

Figure 3-2. Directory listing using /P qualifier

```
KEYB        COM    9041    7-24-87  12:00a
LABEL       COM    2346    7-24-87  12:00a
MODE        COM    15440   7-24-87  12:00a
MORE        COM    282     7-24-87  12:00a
NLSFUNC     EXE    3029    7-24-87  12:00a
PRINT       COM    8995    7-24-87  12:00a
RECOVER     COM    4268    7-24-87  12:00a
SELECT      COM    4132    7-24-87  12:00a
SORT        EXE    1946    7-24-87  12:00a
SUBST       EXE    10552   7-24-87  12:00a
SYS         COM    4725    7-24-87  12:00a
          34 File(s)    5120 bytes free
```

Figure 3-3. Resumption of directory listing after pressing a key

Tip: *If you know the name of the file that you want, simply include the file name in the DIR command line.*

By default, DIR displays information on all of the files on your disk. If you are interested in a specific file—DISKCOPY for example—you can specify the file name in the DIR command line.

```
Volume in drive A is MS330PP01
Directory of A:\

4201        CPI    5202        CPI    ANSI        SYS    APPEND    EXE  ASSIGN    COM
ATTRIB      EXE    CHKDSK      COM    COMMAND     COM    COMP      COM  COUNTRY   SYS
DISKCOMP    COM    DISKCOPY    COM    DISPLAY     SYS    DRIVER    SYS  EDLIN     COM
EXE2BIN     EXE    FASTOPEN    EXE    FDISK       COM    FIND      EXE  FORMAT    COM
GRAFTABL    COM    GRAPHICS    COM    JOIN        EXE    KEYB      COM  LABEL     COM
MODE        COM    MORE        COM    NLSFUNC     EXE    PRINT     COM  RECOVER   COM
SELECT      COM    SORT        EXE    SUBST       EXE    SYS       COM
     34 File(s)    5120 bytes free
```

Figure 3-4. Directory listing using the /W qualifier

```
A> DIR  DISKCOPY

Volume in drive A is MS330PP01
Directory of  A:\

DISKCOPY      COM          6264     7-24-87  12:00a
            1 File(s)      5120 bytes free
```

In the same manner, DIR allows you to include the file's extension in the command line.

```
A> DIR  DISKCOPY.COM

Volume in drive A is MS330PP01
Directory of  A:\

DISKCOPY      COM          6264     7-24-87  12:00a
            1 File(s)      5120 bytes free
```

Tip: *If you want to list files contained on a disk other than the default, simply include a disk drive specifier in the DIR command line.*

If your computer has two floppy-disk drives, insert your DOS Supplemental Programs Disk in drive B and issue the following directory command:

```
A> DIR  B:

Volume in drive B is MS330PP02
Directory of  B:\

BACKUP      COM      29976      7-24-87  12:00a
DEBUG       COM      15866      7-24-87  12:00a
EGA         CPI      49065      7-24-87  12:00a
GWBASIC     EXE      80592      7-24-87  12:00a
KEYBOARD    SYS      19735      7-24-87  12:00a
LCD         CPI      10752      7-24-87  12:00a
LINK        EXE      39172      7-24-87  12:00a
PRINTER     SYS      13559      7-24-87  12:00a
RAMDRIVE    SYS      6481       7-24-87  12:00a
```

```
REPLACE     EXE     13234   7-24-87  12:00a
RESTORE     COM     35650   7-24-87  12:00a
SHARE       EXE     8608    7-24-87  12:00a
TREE        COM     3540    7-24-87  12:00a
XCOPY       EXE     11216   7-24-87  12:00a
FC          EXE     15974   7-24-87  12:00a
        15 File(s)    2048 bytes free
```

As you can see, DIR listed the files contained on the disk in drive B.

Tip: *The first two lines of the directory listing reveal the disk drive containing the files displayed.*

```
A> DIR

Volume in drive A is MS330PP02
Directory of  A:\
```

This directory listing is of the files contained on the disk in drive A.

Tip: *If you are interested in a specific file that resides on a disk other than the current default, simply precede the file name with a disk drive specifier.*

Just as DIR allows you to list the files contained on a disk other than the default by simply placing a disk drive specifier in your command line, for example

```
A> DIR  B:
```

it also allows you to display the directory listing for a specific file that resides on a disk other than the default, as shown here:

```
A> DIR  B:FILENAME.EXT
```

Trap: *If DIR cannot find the file name you specified in the command line, it will display the following message:*

```
File not found
```

Do not confuse this error message with the message

```
Bad command or file name
```

which DOS displays when it cannot find a command on the disk. The message

```
File not found
```

tells you that DOS successfully executed the command that you entered, but failed to find the file specified in the command line.

Naming Files

As you have seen, DOS file names are comprised of a file name and an optional extension. The file name must be from one to eight characters in length. All of the characters in the alphabet are valid in DOS file names along with these characters:

~!@#$%^&*()-_{}'

Tip: *Give your files meaningful names.*

One of the most important steps that you can take when working with DOS is to assign meaningful names to your files.

Although a simple file name such as X.DAT may seem sufficient when you create a file, compare it to the file name EXPENSES.MAY. Several days from

now when you are searching your directory for a specific file, meaningful file names will make your search much easier.

Since you have complete control over the names of your files, make them meaningful. By using 8-character names and complete file extensions, your file names can become quite meaningful, as shown here:

JUNE.EXP	TAXES.88	MATH.NTS
MEETINGS.MON	MEETINGS.TUE	PHONE.###
GROCERY.LST	BIRTHDAY.!!!	WORK.PAY

Using Wildcard Characters

Many of the DOS commands that will be examined throughout the remainder of this book allow you to apply the command to a group of files by using two characters, the asterisk (*) and question mark (?). These are called the DOS *wildcard characters.*

These two characters allow DOS to match file names by ignoring the character in that position in the file. Using the DOS DIR command, for example, you can direct DOS to display all the files that start with the letters "CHAP" and also have the extension TXT.

```
A> DIR  CHAP????.TXT
```

A disk containing chapters of this book might display

```
A> DIR  CHAP????.TXT

Volume in drive A is DOSBOOK
Directory of  A:\

CHAP4        TXT     23040   6-16-88  1:31p
CHAP3        TXT     24960   6-16-88  8:53p
          2 File(s)  302912 bytes free
```

Notice that because DOS file names can be eight characters in length, the previous command filled the last four characters of the file name with the ques-

tion mark wildcard character. Assuming that your DOS Program Disk is in drive A, the command

```
A> DIR  DISKCO??.COM
```

will display

```
A> DIR  DISKCO??.COM

Volume in drive A is MS330PP01
Directory of  A:\

DISKCOMP      COM          5848     7-24-87  12:00a
DISKCOPY      COM          6264     7-24-87  12:00a
            2 File(s)     5120 bytes free
```

As you can see, DOS ignored the last two characters in the file name. Likewise, to list all of the files on your disk, you can use all wildcard characters.

```
A> DIR  ????????.???
```

Remember, though, that by default DIR lists all of the files on your disk.

A similar, but more powerful, wildcard character is the asterisk (*). Unlike the question mark, which directs DOS to ignore the character in the position specified, the asterisk directs DOS not only to ignore the character in the position containing the asterisk, but also all of the characters that follow.

Using this wildcard character you can change the previous command from

```
A> DIR  CHAP????.TXT
```

to

```
A> DIR  CHAP*.TXT
```

In both cases, DOS will display

```
A> DIR  CHAP*.TXT

Volume in drive A is DOSBOOK
Directory of  A:\
```

```
CHAP4        TXT    23040   6-16-88  1:31p
CHAP3        TXT    24960   6-16-88  8:53p
        2 File(s)        302912 bytes free
```

Likewise, again using your DOS Program Disk, the command

```
A> DIR  DISKCO*.COM
```

will display

```
A> DIR  DISKCO*.COM

Volume in drive A is MS330PP01
Directory of  A:\

DISKCOMP     COM        584      7-24-87  12:00a
DISKCOPY     COM        6264     7-24-87  12:00a
        2 File(s)      5120 bytes free
```

Using the asterisk wildcard character, you can abbreviate the file name to an even greater extent with the command

```
A> DIR  DISK*.COM

Volume in drive A is MS330PP01
Directory of  A:\

DISKCOMP     COM        584      7-24-87  12:00a
DISKCOPY     COM        6264     7-24-87  12:00a
        2 File(s)      5120 bytes free
```

To view all of the files of a specific type, you can use the asterisk wildcard character followed by an extension.

```
A> DIR *.SYS

Volume in drive A is MS330PP01
Directory of A:\

ANSI        SYS        1647    7-24-87  12:00a
COUNTRY     SYS        11254   7-24-87  12:00a
DISPLAY     SYS        1125    7-24-87  12:00a
DRIVER      SYS        1165    7-24-87  12:00a
          4 File(s)    5120 bytes free
```

If you use the asterisk for both the file name and extension, as follows

```
A> DIR *.*
```

DIR will list all of the files on your disk.

Tip: *To direct DOS to group files while ignoring a single letter in a specific position of a file name, use the question mark wildcard character: DISKCO??.COM.*

Tip: *To direct DOS to group files while ignoring a single letter in a specific position of a file name along with all the characters that follow the position specified, use the following asterisk wildcard character: DISK*.COM.*

You can use the DOS wildcard characters for files that reside on disks other than the default disk, by simply using a disk drive specifier, as shown here:

```
A> DIR B:*.COM
```

In this case, DIR will display all of the files on drive B that have the extension COM. If no such files exist, DOS will display the message

```
File not found
```

Making Duplicate Copies of Your Files

Quite possibly the most commonly used DOS command is COPY, which allows you to copy the contents of one DOS file to another. As you will see in this section, the duplicate file can reside on the same disk with a different name from the original, or on a different disk.

To start, copy the file DISKCOPY.COM from the DOS disk in drive A to a new disk in drive B.

Tip: *To copy a file to a disk other than the current default, simply precede the name of the file being created (the target file) with a disk drive specifier.*

Given the command

```
A> COPY DISKCOPY.COM B:DISKCOPY.COM
        1 File(s) copied

A>
```

DOS will copy the contents of the file DISKCOPY.COM to drive B, assigning the file name DISKCOPY.COM to the duplicate copy. A directory listing of the disk in drive B reveals

```
A> DIR B:

Volume in drive B has no label
Directory of B:\

DISKCOPY     COM        6264    7-24-87  12:00a
        1 File(s)    355328 bytes free
```

If you want to assign a different name to the target file, such as DCOPY.COM, simply specify the desired name, as shown here:

```
A> COPY DISKCOPY.COM B:DCOPY.COM
        1 File(s) copied

A>
```

A directory of the disk in drive B now shows both files.

```
A> DIR B:

Volume in drive B has no label
Directory of B:\

DISKCOPY    COM     626     7-24-87  12:00a
DCOPY       COM     6264    7-24-87  12:00a
        2 File(s)   348160 bytes free
```

If you do not specify the disk drive letter in front of the target file name

```
A> COPY DISKCOPY.COM DCOPY.COM
```

DOS will copy the contents of DISKCOPY.COM to a file named DCOPY.COM on the same disk.

Trap: *Warning! The DOS COPY command overwrites any file with the same name as the target file name in the COPY command line.*

Be careful when you use the DOS COPY command. If a file with the same name as the target file you specify already exists, COPY will overwrite the existing file without displaying an error message or warning.

Trap: *DOS will not allow you to copy a file to a target file with the same name on the same disk.*

If DOS displays the error message

```
File cannot be copied onto itself
        0 File(s) copied
```

it is telling you that the target name of your file copy must be different from the source file name. If this error occurs, you have probably simply omitted a disk drive specifier for the target file.

If you use DOS wildcard characters in the target file name of a COPY operation, DOS will copy the file to a new destination using the same name as the source file. For example, given the command

```
A> COPY DISKCOPY.COM B:*.*
```

DOS will copy the file DISKCOPY.COM to drive B exactly as if you had entered the command

```
A> COPY DISKCOPY.COM B:DISKCOPY.COM
```

Trap: *If you omit the extension in the target file name during a COPY operation, so will DOS.*

Remember that the 3-character file name extension is optional. If you don't specify the extension on a file copy operation, for example

```
A> COPY DISKCOPY.COM DISKCOPY
```

DOS will omit the extension on the target file, naming it DISKCOPY as you specified.

Tip: *If you need to copy multiple files to a new disk, use the DOS wildcard characters if possible.*

Just as DOS allows you to copy a single file, by using DOS wildcard characters as shown here, you can copy multiple files.

```
A> COPY *.SYS B:*.SYS
```

Assuming that your DOS Program Disk is in drive A, DOS will copy all files with the SYS extension to drive B, assigning each the same name as its source file. To copy all of the files on the disk in drive A to B, use the command

```
A> COPY *.* B:*.*
```

Tip: *If you want to copy one or more files from one disk to another without changing the file names, you can specify a disk drive letter as the target destination of the file copy.*

Remember that the DOS command

```
A> COPY DISKCOPY.COM B:DISKCOPY.COM
```

copies the contents of the file DISKCOPY.COM to a file on drive B, assigning the same name to the newly created file.

You also learned that you can use wildcard characters to achieve the same result.

```
A> COPY DISKCOPY.COM B:*.*
```

To reduce your keystrokes even more, DOS allows you to specify simply the target drive when the target file name is not changed, as shown here:

```
A> COPY DISKCOPY.COM B:
```

Tip: *If the source file of a copy operation resides on a disk other than the current default, specify the disk drive letter before the file name.*

If your current disk drive is drive A and you need to copy a file that resides on drive B, simply precede the file name with a disk drive specifier, as shown here:

```
A> COPY B:FILENAME.EXT FILENAME.EXT
```

You can use DOS wildcard characters to achieve the same result:

```
A> COPY B:FILENAME.EXT *.*
```

If you do not want to change the name of the target file, you can omit the target name completely (assuming that the source file resides on a different disk).

```
A> COPY B:FILENAME.EXT
```

The following pairs of commands are identical.

A> COPY *.* B:*.*	⟷	A> COPY *.* B:
A> COPY FILENAME.EXT B:FILENAME.EXT	⟷	A> COPY FILENAME.EXT B:
A> COPY FILENAME.EXT B:FILENAME.EXT	⟷	A> COPY FILENAME.EXT B:*.*
A> COPY B:FILENAME.EXT FILENAME.EXT	⟷	A> COPY B:FILENAME.EXT *.*
A> COPY B:FILENAME.EXT FILENAME.EXT	⟷	A> COPY B:FILENAME.EXT

Trap: *If your target disk does not have enough space available for the file, DOS will display the following error message.*

```
Insufficient disk space
```

Just as the files in your office cabinets are restricted in the amount of material they can hold, so are your disks. Remember that the last piece of in-

formation displayed by the DOS DIR command is the number of available bytes of disk space, as shown here:

```
55 File(s)    7824 bytes free
```

If the file you are trying to copy exceeds this amount, DOS will be unable to copy the file to the disk.

Trap: *Unless you use DOS subdirectories, your disks are restricted to a limited number of files.*

Standard 5 1/4-inch floppy disks allow you to place 112 files on your disk, regardless of the number of bytes of available disk space. If you attempt to create a 113th file, DOS will display the message

```
File creation error
```

In later chapters you will learn how DOS subdirectories allow you to get around this restriction.

Changing a File's Name

Just as the DOS COPY command allows you to duplicate a file, the DOS RENAME command allows you to change a file's name.

Tip: *You can abbreviate the RENAME command to REN.*

As you will find in later chapters, DOS allows you to abbreviate a few of the most commonly used commands. In the case of RENAME, you can use either the complete command name (RENAME) or the letters REN.

The command

```
A> RENAME CHAP1.TXT CHAP1.SSS
```

changes the name of the file CHAP1.TXT to CHAP1.SSS. This command is functionally equivalent to

```
A> REN CHAP1.TXT CHAP1.SSS
```

If DOS cannot find the file to rename, it will display the message

```
Duplicate file name or File not found
```

If this occurs, simply perform a directory listing to verify the file's name.

Trap: *DOS does not allow you to rename a file to a name that already exists on that disk.*

To protect you from overwriting an existing file, the DOS RENAME command forces you to rename your files to a name that does not already exist on that disk. If you use an existing file name as your target file, DOS will display the message

```
Duplicate file name or File not found
```

If you want to overwrite an existing file, you must first delete the existing file (using the DEL command discussed next) and then perform the RENAME command.

Trap: *DOS does not allow you to rename a file from one disk to another.*

The DOS RENAME command does not move files; it simply renames them. If you attempt to rename a file from one disk drive to another, the command will fail and DOS will display the message

```
Invalid parameter
```

Likewise, if you specify a disk drive letter for your target file without changing disk drives, for example

```
A> REN  B:OLDNAME.EXT  B:NEWNAME.EXT
```

the command will still fail.

```
A> REN  B:OLDNAME.EXT  B:NEWNAME.EXT
Invalid parameter

A>
```

The correct command in this case is

```
A> REN  B:OLDNAME.EXT  NEWNAME.EXT
```

DOS will change the name of the file on drive B as desired.

If you needed to rename and move a file to the disk in drive B, you could do so by first copying it to the disk in drive B with the name desired, and then delete it from the disk in drive A.

Trap: *RENAME cannot change the names of read-only files.*

Later in this book you will learn how to protect your files by marking them as read-only. If you later attempt to modify or delete a read-only file, DOS will display the message

```
Access Denied
```

Trap: *RENAME cannot rename a file that resides on a write-protected disk.*

As stated in Chapter 1, DOS cannot modify the contents of a write-protected disk. If you attempt to change the name of a file that resides on such a disk, the command will fail and DOS will display the message

```
Write protect error writing drive A
Abort, Retry, Fail?
```

Tip: *If you need to rename several files in a consistent manner, use the DOS wildcard characters.*

Just as the DOS COPY command fully supports wildcard characters, so too does RENAME. For example, the following command renames all of the files that begin with the letters "CHAP" and have the extension TXT to files with the extension SSS.

```
A> REN CHAP*.TXT *.SSS
```

Deleting Files

Throughout your sessions with your computer, you will create and modify many files. When you no longer need particular files, the DOS DEL command allows you to remove them from your disk. DEL is an internal DOS command.

Trap: *Once you delete a file, DOS does not provide you with a means of getting the file back.*

The DOS DEL command erases files from your disk permanently. Therefore, use this command with caution. There are some third-party software packages that can "undelete" files, but they are not 100 percent successful.

To delete a file that resides on the current disk drive, type **DEL** and the name of the file.

```
A> DEL FILENAME.EXT
```

Just as with other DOS commands, DEL allows you to manipulate files that reside on disks other than the current drive by specifying a disk drive letter before the file name, as shown here:

```
A> DEL B:FILENAME.EXT
```

Tip: *DEL cannot affect files on a write-protected disk.*

DOS cannot modify the contents of a write-protected disk. If you do not anticipate the files on your disk changing (as is the case with your original disks), write protect the disk as discussed in Chapter 1.

Trap: *DEL cannot delete a read-only file.*

Later in this book you will learn how to protect your files by marking them as read-only. If you attempt to delete a read-only file, DOS will display the message

```
Access Denied
```

Tip: *If you need to delete multiple files, use the DOS wildcard characters; however, do so with caution.*

Be careful when you use DOS wildcard characters in conjunction with the DEL command. An errant command can have a devastating impact on your files. The following command deletes all the files on the current disk that have the extension BAK.

```
A> DEL *.BAK
```

In a similar manner, the command

```
A> DEL *.*
```

directs DOS to delete all of the files from your disk. Because this command can have a catastrophic effect on your disk if you enter it in error, DOS will first prompt you with

```
Are you sure (Y/N)?
```

If you want DOS to delete all the files on your disk, type **Y** and press ENTER. To cancel the command, simply type **N** and press ENTER.

You will most likely use the DOS COPY, RENAME, and DEL commands on a daily basis, so it is worth taking time now to understand them.

Tips and Traps Summary

Looking at Directories

Tip: The DOS DIR command allows you to display the names of all the files that reside on your disk.

Tip: DIR displays a file's name, extension, size, and creation/modification date and time.

Tip: DOS file names are actually comprised of two parts, a file name and an optional extension.

Tip: A file's extension is a good guide to the file's contents.

Tip: DOS displays each file's size in bytes. A byte is essentially one character.

Tip: The date and time displayed in the DOS directory listing are the date and time that the file was either created or last modified.

Trap: If a disk contains many files, its directory listing may scroll quickly on the screen, causing you to miss your desired file.

Tip: DIR provides the /P qualifier, which suspends scrolling of file names with each screenful of information.

Tip: If you are not interested in the file size, date, and time fields, you can use the /W qualifier, which directs DIR to suppress the display of those fields.

Tip: If you know the name of the file that you want, simply include the file name in the DIR command line.

Tip: If you want to list files contained on a disk other than the default, simply include a disk drive specifier in the DIR command line.

Tip: The first two lines of the directory listing reveal the disk drive containing the files displayed.

Tips and Traps Summary (*continued*)

Tip: If you are interested in a specific file that resides on a disk other than the current default, simply precede the file name with a disk-drive specifier.

Trap: If DIR cannot find the file name you specified in the command line, it will display the following message

```
File not found
```

Naming Files

Tip: Give your files meaningful names.

Using Wildcard Characters

Tip: To direct DOS to group files while ignoring a single letter in a specific position of a file name, use the question mark wildcard character: DISKCO??.COM.

Tip: To direct DOS to group files while ignoring a single letter in a specific position of a file name along with all the characters that follow the position specified, use the asterisk wildcard character: DISKCO*.COM.

Making Duplicate Copies of Your Files

Tip: To copy a file to a disk other than the current default, simply precede the name of the file being created (the target file) with a disk-drive specifier.

Trap: Warning! The DOS COPY command overwrites any file with the same name as the target file name in the COPY command line.

Tips and Traps Summary (*continued*)

Trap: DOS will not allow you to copy a file to a target file with the same name on the same disk.

Trap: If you omit the extension in the target file name during a COPY operation, so will DOS.

Tip: If you need to copy multiple files to a new disk, use the DOS wildcard characters if possible.

Tip: If you want to copy one or more files from one disk to another without changing the file names, you can specify a disk-drive letter as the target destination of the file copy.

Tip: If the source file of a copy operation resides on a disk other than the current default, specify the disk-drive letter before the file name.

Trap: If your target disk does not have enough space available for the file, DOS will display the following error message.

```
Insufficient disk space
```

Trap: Unless you use DOS subdirectories, your disks are restricted to a limited number of files.

Changing a File's Name

Tip: You can abbreviate the RENAME command to REN.

Trap: DOS does not allow you to rename a file to a name that already exists on that disk.

Trap: DOS does not allow you to rename a file from one disk to another.

Trap: RENAME cannot change the names of read-only files.

Trap: RENAME cannot rename a file that resides on a write-protected disk.

Tips and Traps Summary (*continued*)

Tip: If you need to rename several files in a consistent manner, use the DOS wildcard characters.

Deleting Files

Trap: Once you delete a file, DOS does not provide you with a means of getting the file back.

Tip: DEL cannot affect files on a write-protected disk.

Trap: DEL cannot delete a read-only file.

Tip: If you need to delete multiple files, use the DOS wildcard characters; however, do so with caution.

4

Handling Your Disks

Thus far, most of the commands that you have used worked strictly with your DOS disks. In this chapter you will learn

- How to prepare floppy disks for use by DOS
- The differences between fixed-disk drives (also called hard-disk drives) and floppy-disk drives
- How to prepare your fixed disk so that it can be used to start DOS
- How to check the current status of your disks
- How to assign a name to your disk and later display this name

Just as Chapter 3 taught you how to handle your files, this chapter gets you up and running with your disks.

Handling Your Floppy Disks

Chapter 1 discussed handling your disks. As you work with your disks, keep in mind the rules shown in Figure 4-1.

Preparing a Disk for Use by DOS

When you purchase floppy disks for your computer, you must keep in mind the fact that the original disk manufacturer has no idea of what type of computer (Apple, Commodore, IBM, and so on) that you have. Therefore, the manufacturer does not prepare the disk for use on a specific system. Before you can use your disks, you must prepare, or format, them for use by DOS.

Tip: *The DOS FORMAT command prepares a disk for use by DOS.*

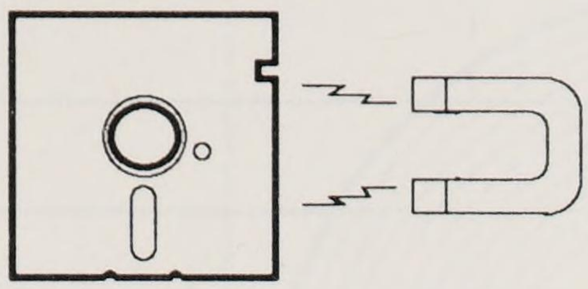
If you perform the following directory listing of your DOS Program Disk

```
A> DIR  FORMAT.COM
```

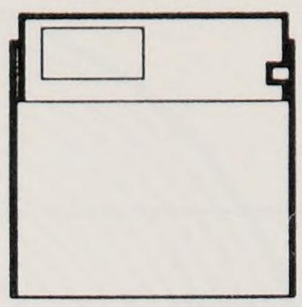
DOS will display

```
A> DIR  FORMAT.COM

Volume in drive A is MS330PP01
Directory of  A:\

FORMAT        COM        11671    7-24-87  12:00a
              1 File(s)     5120 bytes free
```

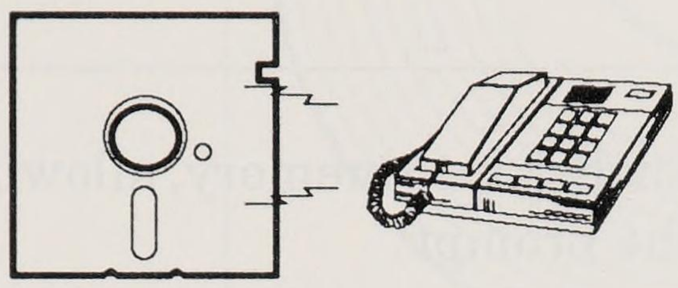
FORMAT is an external DOS command. To invoke FORMAT, your DOS Program Disk must be in one of your disk drives.



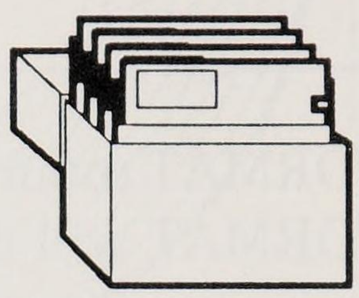
Never place the disk near magnetic devices.



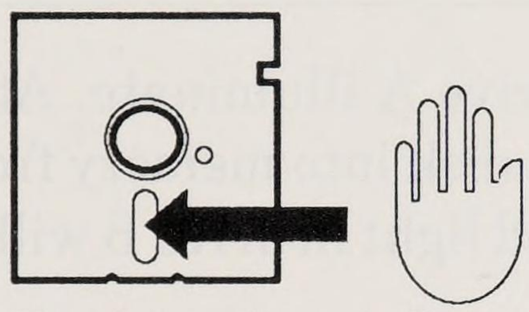
Always place disks back into disk envelopes when you are not using them.



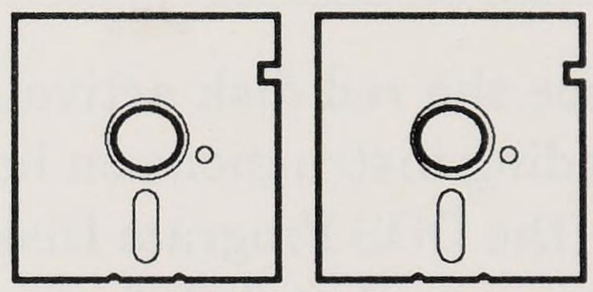
Keep your disk away from your telephone.



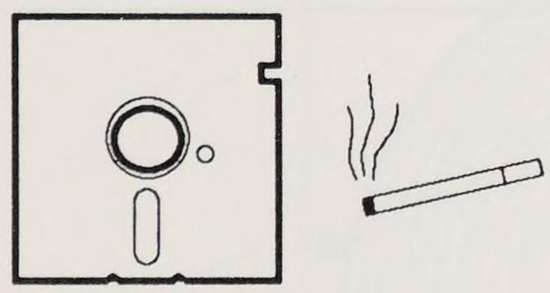
Store your floppy disks in a safe location.



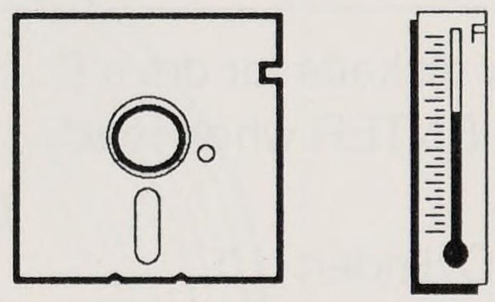
Never touch your floppy-disk media.



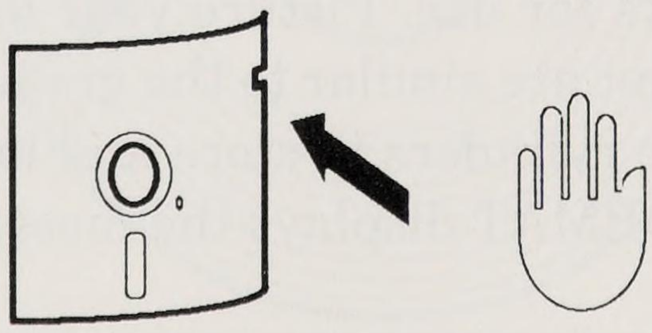
Always make a backup copy of your floppy disk.



Never smoke near floppy disks.



Keep room temperature in the range 50° F to 110 °F.



Never bend floppy disks.

Figure 4-1. Rules for handling floppy disks

Tip: *The simplest and most common way to use FORMAT is as follows:*

```
A> FORMAT B:
```

If you have two floppy-disk drives, place your DOS Program Disk in drive A and a new disk in drive B. Next, issue the command

```
A> FORMAT B:
```

DOS will load the FORMAT command from disk into memory, allowing the command to execute. FORMAT will display the prompt

```
Insert new diskette for drive B:
and strike ENTER when ready
```

You will see the red disk activation light in drive A illuminate. At that time, DOS is loading instructions on how to format a disk into memory from the disk in drive A (the DOS Program Disk). Then the red light in drive B will illuminate as DOS prepares that disk for use.

Once you have inserted your DOS disk in drive B, press the ENTER key. At that time, DOS will display

```
Insert new diskette for drive B:
and strike ENTER when ready

Head:  0 Cylinder:  0
```

DOS is preparing your disks for use. Picture your disk as being composed of several rings or cylinders that are similar to the grooves in a record album (see Figure 4-2). DOS uses these cylinders to store and locate information on both sides of your disk. When FORMAT displays the message

```
Head:  0 Cylinder:  0
```

it is telling you that it is preparing the first cylinder on the first side of your disk for use, as shown in Figure 4-3.

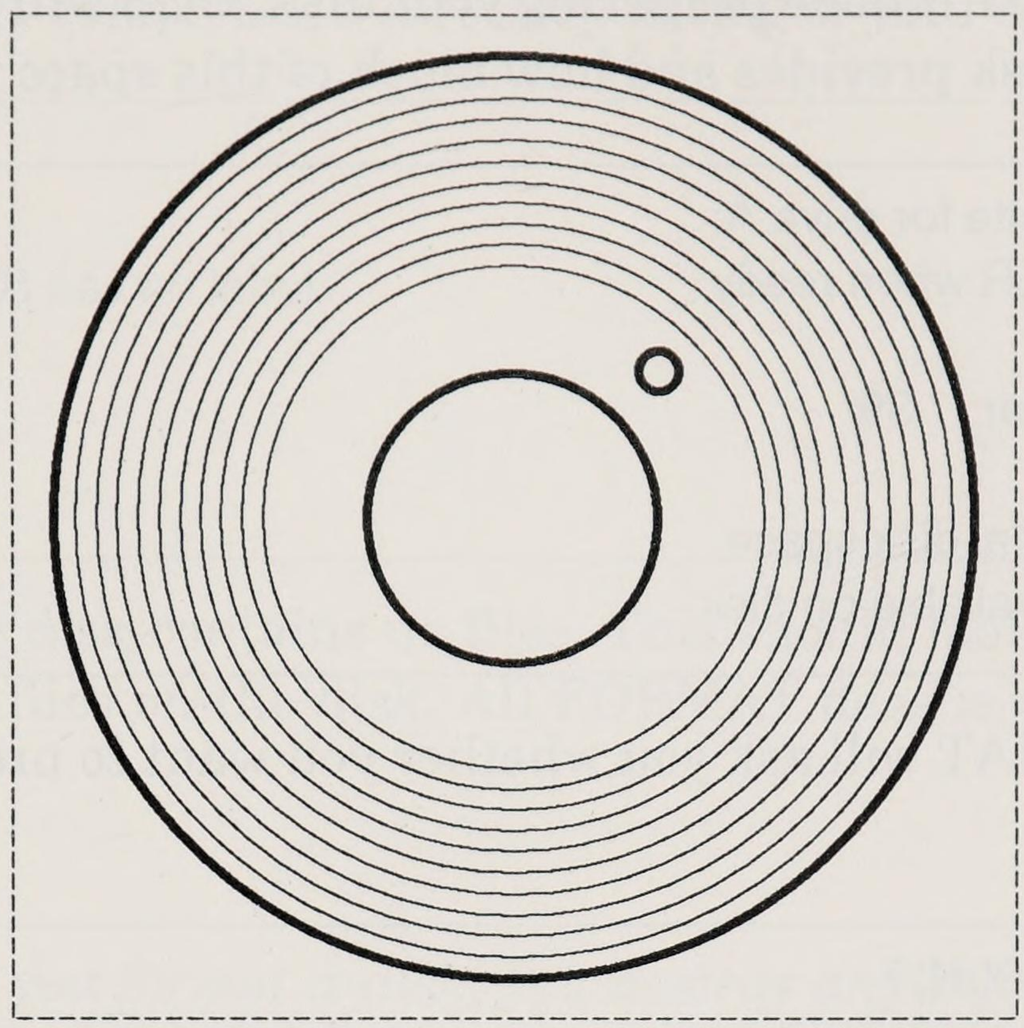


Figure 4-2. Cylinders on floppy disk

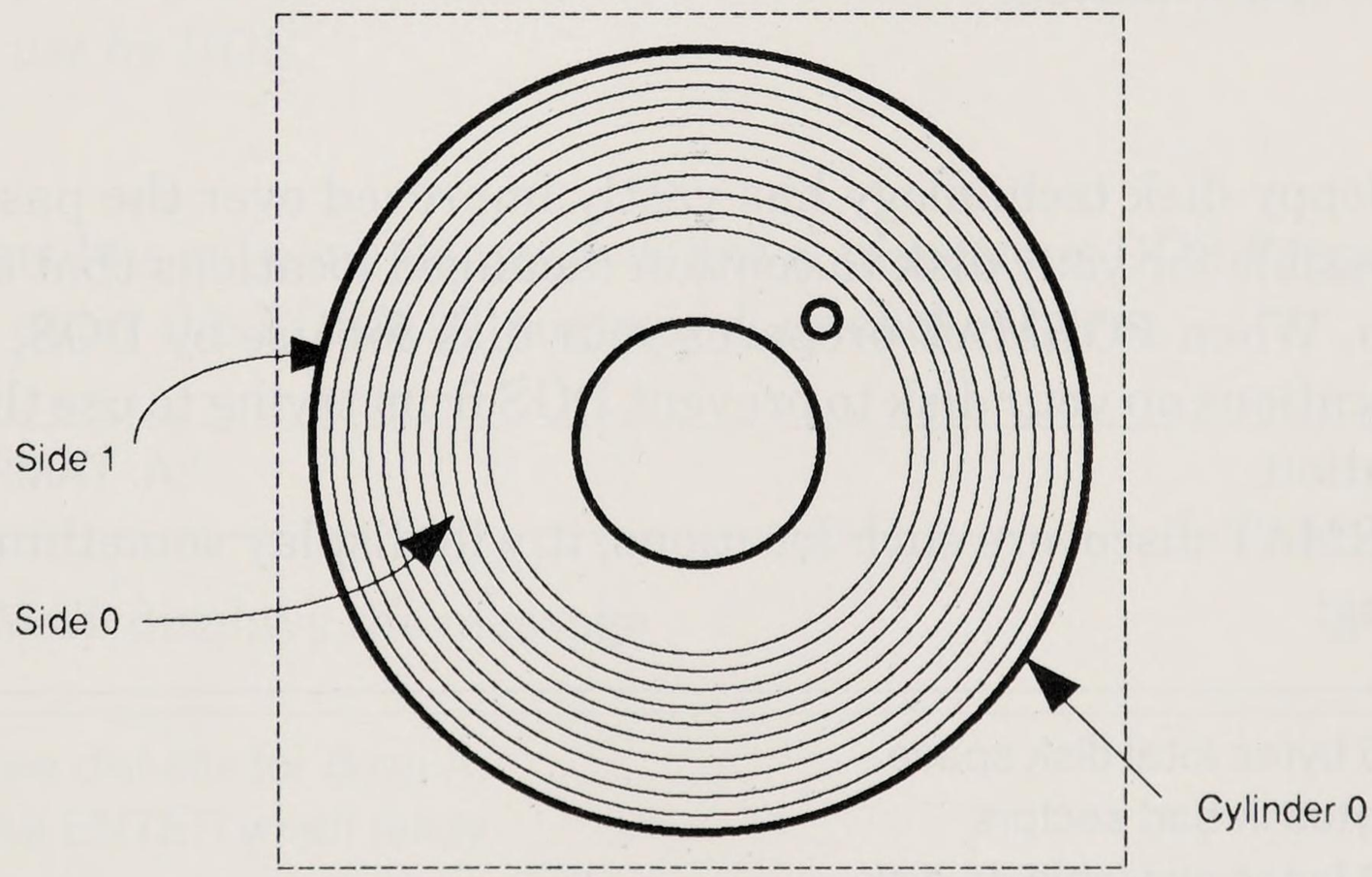


Figure 4-3. First cylinder on first side of disk being formatted

When FORMAT is done preparing your disk, DOS will display the amount of space that the disk provides and how much of this space is usable by DOS.

```
Insert new diskette for drive A:
and strike ENTER when ready

Head:  1 Cylinder:  39

362496 bytes total disk space
362496 bytes available on disk
```

Lastly, FORMAT will ask you whether you want to prepare a second disk for use by DOS.

```
Format another (Y/N)?
```

If you have other disks to prepare, you can do so now by simply typing **Y** and pressing ENTER. If you do not want to format other disks, type **N** and press ENTER.

Trap: *Some disks contain damaged locations (sectors) that cannot store information.*

Although floppy-disk technology has vastly improved over the past few years, it is still possible for your disk to contain damaged locations that cannot store information. When FORMAT prepares your disk for use by DOS, it notes the damaged locations on your disk to prevent DOS from trying to use them for storing information.

If FORMAT discovers such locations, it will display something similar to the following:

```
362496 bytes total disk space
4096 bytes in bad sectors
358400 bytes available on disk
```

If several of your disks contain damaged locations, consider purchasing a different brand of disks.

Once FORMAT completes, a directory listing of your disk displays

```
A> DIR  B:

Volume in drive B has no label
Directory of  B:\

File not found
```

As you can see, the disk contains no files. This should make sense—you have not yet placed any files on the disk. All FORMAT does is prepare the disk to receive files.

Trap: *Once you format a disk, you destroy any information previously stored on it.*

Be very careful with the DOS FORMAT command. If you inadvertently format a disk that contains needed information, the information is lost.

Tip: *FORMAT also allows single-disk drive systems to prepare disks for use by DOS.*

If your system has only one floppy-disk drive, place your DOS Program Disk in drive A and enter the FORMAT command, as shown here:

```
A> FORMAT  A:
```

When FORMAT displays the message

```
Insert new diskette for drive A:
and strike ENTER when ready
```

simply remove your DOS Program Disk from drive A and insert your new disk. When you press ENTER, FORMAT will begin preparing your disk for use as shown previously.

Trap: *Just because a disk has been formatted for use under DOS does not mean that the disk can start (boot) DOS.*

Remember from Chapter 1 that DOS requires several special files on disk before it can boot. To verify this, place your newly formatted disk in drive A and press the CTRL, ALT, and DEL key combination to restart DOS.

As always, you will first see the disk drive activation light for drive A illuminate as your computer searches for DOS on that disk. Since the files required to reboot DOS do not reside on the disk, DOS cannot start, and your computer will display

```
Non-System disk or disk error
Replace and strike any key when ready
```

Insert your DOS Program Disk in drive A and press any key to continue.

Tip: *To create a disk that can start DOS, you must use the /S qualifier in the FORMAT command.*

Just as FORMAT allows you to prepare your disks for use by DOS, it also allows you to create a disk that can start (boot) DOS. To do so, simply add the /S qualifier to the FORMAT command, as shown here:

```
A> FORMAT B: /S
```

FORMAT will prepare each cylinder on your disk for use by DOS. It will then copy to the disk the files that DOS requires to boot. Here is the display:

```
A> FORMAT B: /S

Insert new diskette for drive B:
and strike ENTER when ready

Head: 1 Cylinder: 39

System transferred
```

```
362496 bytes total disk space
78848 bytes used by system
283648 bytes available on disk

Format another (Y/N)?
```

The DOS system files consume a considerable amount of disk space.

```
362496 bytes total disk space
78848 bytes used by system        ←—— Space consumed on disk by DOS files
283648 bytes available on disk
```

A directory listing of this disk now reveals

```
A> DIR  B:

Volume in drive B has no label
Directory of  B:\

COMMAND      COM          25276    7-24-87  12:00a
            1 File(s)     283648 bytes free
```

The COMMAND.COM file contains your DOS internal commands and is responsible for displaying the DOS prompt. In order for a disk to be able to start DOS, the COMMAND.COM file must be present on the disk.

Trap: *Not all versions of DOS copy the COMMAND.COM file to the target disk during formatting.*

Upon completion of the DOS FORMAT /S command, if your target disk does not contain the COMMAND.COM file you will see this display:

```
A> DIR  B:

Volume in drive B has no label
Directory of  B:\

File not found
```

You must copy COMMAND.COM from your DOS Program Disk to the new disk, as shown here:

```
A> COPY COMMAND.COM B:
```

If this file does not reside on your disk, your computer will display

```
Bad or missing Command Interpreter
```

when you attempt to boot DOS from this disk.

Tip: *Do not format all of your floppy disks as bootable.*

As you have just seen, the files that DOS requires on your disk in order for the disk to boot consume a considerable amount of disk space. Most of you will normally start DOS by using your DOS Program Disks. Therefore, you don't need a collection of bootable disks. If you examine the information displayed in the previous FORMAT command, you can see that the DOS system files consume about one-sixth of your disk's storage capacity. This space is no longer available to store information. If you never intend to use a particular disk to boot DOS, you are wasting disk space if you place the DOS system files on that disk.

Trap: *Not all DOS floppy disks are readily interchangeable.*

Most personal computers have a "standard" 360K floppy-disk drive, which supports double-sided, double-density disks. However, the IBM PC/AT and AT compatibles often have a disk drive called a 1.2MB drive. This disk drive uses disks called quad-density, which allow you to store about four times as much information as a standard 360K disk.

Although both of these disks are fully functional under DOS, a problem arises when you have to exchange files between a 1.2MB disk and a 360K disk. If you have a 1.2MB disk drive and you must prepare a disk to be read by a 360K drive, the FORMAT command allows you to do so.

Tip: *The FORMAT command with a /4 qualifier allows you to format a 360K disk in your 1.2MB disk drive.*

A 360K disk drive can only read 360K disks. To create a 360K disk in your 1.2MB disk drive, you must use the /4 qualifier of the FORMAT command, as shown here:

```
A> FORMAT /4
```

The only time you will need this qualifier is when you are exchanging files between a 1.2MB disk drive and a 360K disk drive.

Naming Your Disks

When you get a directory listing for your DOS Program Disk, the first line of the listing displays the name of the disk that you are examining.

```
A> DIR

Volume in drive A is MS330PP01
Directory of  A:\
```

If you examine a newly formatted disk, DOS displays

```
A> DIR  B:

Volume in drive B has no label
Directory of  B:\
```

As you can see, this second disk does not yet have a name assigned to it.

Tip: *Assigning unique names to your disks helps you improve your disk organization.*

Assume, for example, that each year you track tax information on a floppy disk.

```
A> DIR

Volume in drive A has no label
Directory of  A:\

INCOME      JAN      73      6-17-88 3:43a
DEDUCT      JAN      34      6-17-88 3:43a
EXPENSE     JAN      70      6-17-88 3:43a
INCOME      FEB      73      6-17-88  3:43a
DEDUCT      FEB      34      6-17-88  3:43a
EXPENSE     FEB      70      6-17-88  3:43a
          6 File(s)     303104 bytes free
```

At the beginning of the following year, you create a second disk to store that year's tax information. By the end of the third year, you have collected three disks, as shown in Figure 4-4.

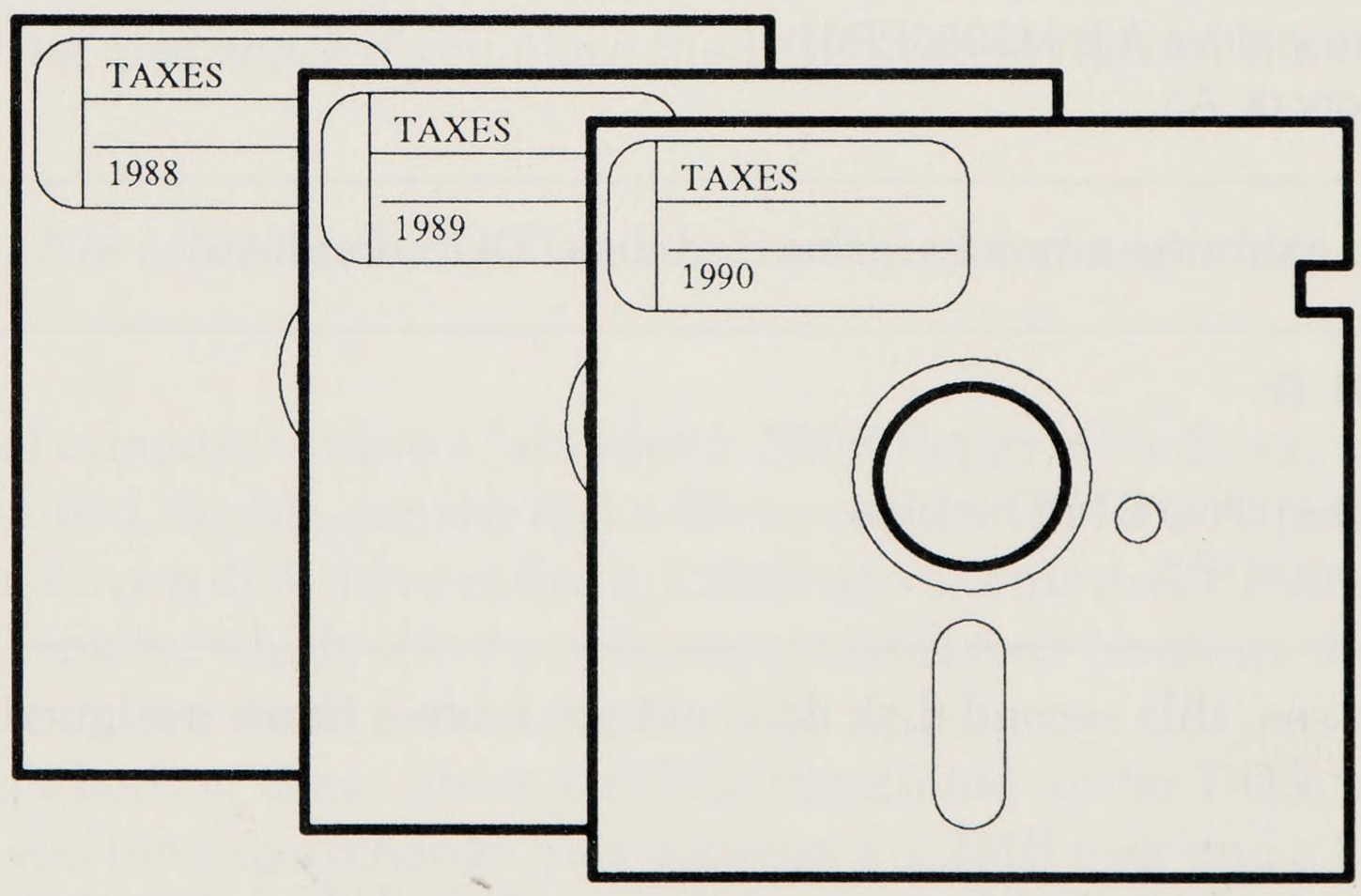


Figure 4-4. Three disks containing tax information

Although these disks are clearly labeled, when you insert one into the computer, you have no easy method of determining if you have placed the correct disk in the drive. To overcome this limitation, the DOS LABEL command allows you to assign a name to your disks.

Tip: *The DOS LABEL command allows you to assign an 11-character name to your disks.*

With your DOS Program Disk in drive A, give the command

```
A> DIR  LABEL.COM

Volume in drive A is MS330PP01
Directory of  A:\

LABEL          COM          2346     7-24-87  12:00a
          1 File(s)                             5120 bytes free
```

LABEL is an external DOS command. To assign a name to the floppy disk in drive B, for example, simply issue the command

```
A> LABEL  B:
```

LABEL will respond by prompting you to enter a volume (or disk) name.

```
A> LABEL  B:

Volume in drive B has no label

Volume label (11 characters, ENTER for none)?
```

All of the characters that are valid in DOS file names are also valid in disk names. In this example, the disk is named TESTDISK:

```
A> LABEL  B:

Volume in drive B has no label

Volume label (11 characters, ENTER for none)? TESTDISK
```

Trap: *If the volume name that you enter is invalid, LABEL will display an error message and prompt you to enter a new label.*

Remember that label names can contain up to 11 characters. All of the characters valid for file names can be used. A directory listing of the disk in drive B now reveals

```
Volume in drive B is TESTDISK
Directory of B:\
```

Tip: *LABEL allows you to specify the disk label in the LABEL command line.*

To change the name of a disk without having to respond to the LABEL prompt, simply type in the desired disk name at the DOS prompt, as shown here:

```
A> LABEL B:TESTDISK
```

If the label that you enter is valid, LABEL will change the name of the disk as specified and redisplay the DOS prompt:

```
A> LABEL B:TESTDISK
A>
```

In the previous scenario of three floppy disks containing income tax information, one disk could be labeled, as shown here:

```
A> LABEL B:TAXES-1989
```

A directory listing of that disk would reveal

```
A> DIR B:

Volume in drive B is TAXES-1989
Directory of B:\
```

```
INCOME     JAN    73     6-17-88  3:43a
DEDUCT     JAN    34     6-17-88  3:43a
EXPENSE    JAN    70     6-17-88  3:43a
INCOME     FEB    73     6-17-88  3:43a
DEDUCT     FEB    34     6-17-88  3:43a
EXPENSE    FEB    70     6-17-88  3:43a
        6 File(s)      303104 bytes free
```

Tip: *To delete a disk volume label, simply press* ENTER *when LABEL prompts you for a volume name.*

If you press ENTER at the LABEL prompt for a volume name, LABEL will then ask you whether you want to delete the current volume label:

```
Delete current volume label (Y/N)?
```

To delete the current volume label, simply type **Y** and press ENTER.

Tip: *The VOL command displays the disk volume label for the disk drive you specify.*

VOL is an internal DOS command that directs DOS to display the name of the disk in the specified drive. If you simply invoke VOL as

```
A> VOL
```

DOS will display the volume name for the disk in the current drive. If you want to display the volume name of a disk in a drive other than the current drive, simply include a disk drive specifier in the command line:

```
A> VOL  B:
```

For the previously named disk, VOL would display

```
Volume in drive B is TAXES-1989
Directory of  B:\
```

If the disk in the specified drive does not have a name, VOL will display

```
A> VOL B:

Volume in drive B has no label
```

Fixed Disks Versus Floppies

So far, this chapter has focused on floppy disks. However, many DOS users have fixed-disk drives on their systems (see Figure 4-5). Fixed disks are much faster and can store a great deal more information than floppy disks. Table 4-1 compares the number of floppy disks it takes to store the same amount of information available on a fixed disk.

Unlike floppy disks, fixed disks are permanent fixtures in your computer. They can store tremendous amounts of information and access that information quickly.

As you will see in later chapters, DOS treats files that reside on fixed disks just as it treats files on floppy disks. All commands work identically regardless of whether you are using a fixed- or floppy-disk drive. However, fixed-disk users have an extra drive, drive C, available.

Preparing a Fixed Disk For Use by DOS

Depending upon the amount of work that your computer retailer performed for you, the number of steps you need to perform in order to start DOS from your fixed disk will vary. If your system has a fixed disk installed, issue the command

```
A> DIR C:
```

Trap: *If DOS has never been told that your computer has a fixed disk, it will display an error message each time you attempt to access the disk.*

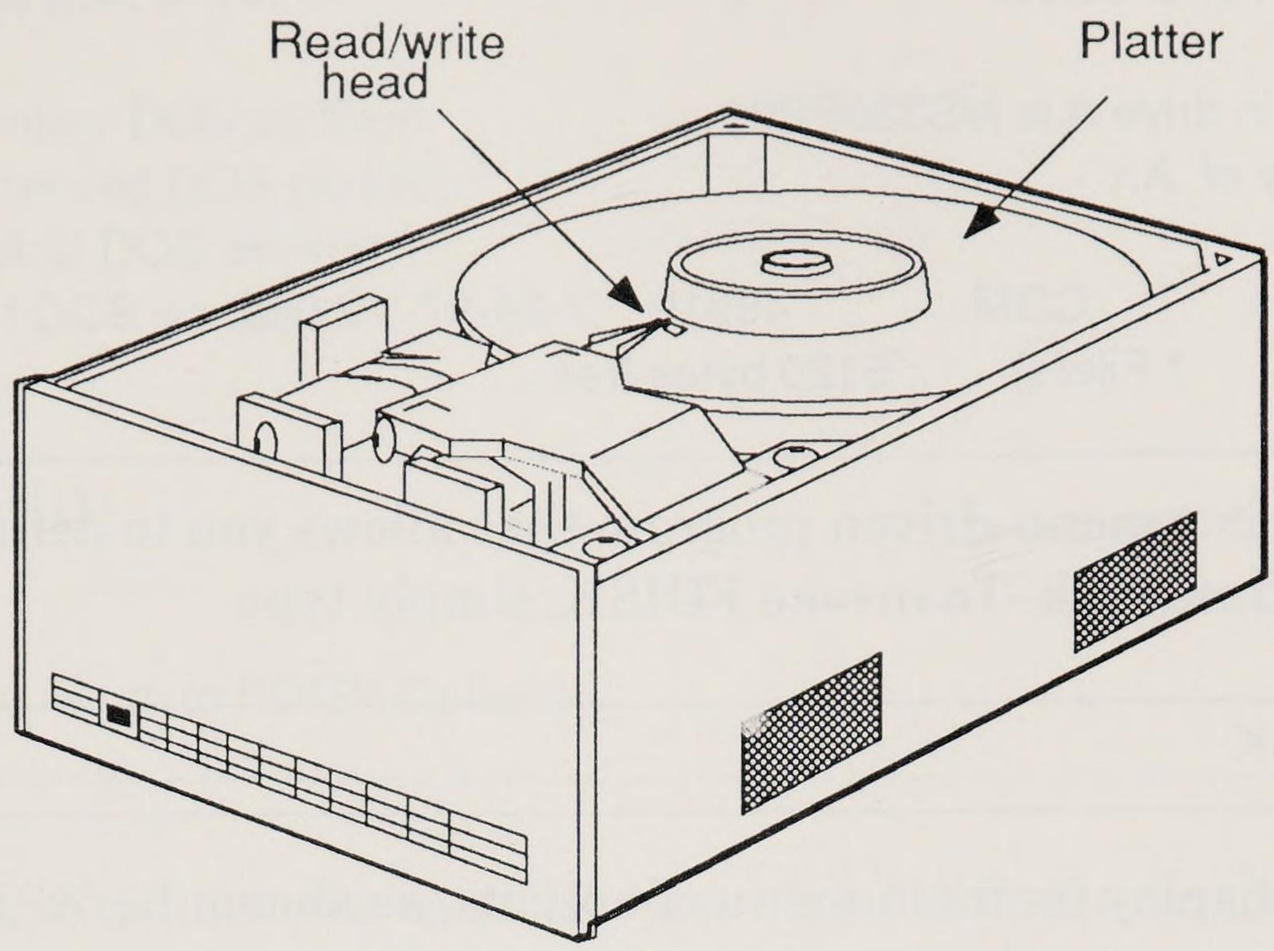


Figure 4-5. Fixed disk

```
A> DIR C:
Invalid drive specification
```

If DOS displays this message for your system, you must invoke the DOS FDISK command. The FDISK.COM file is located on either the DOS Program Disk or the Supplemental Programs Disk.

Table 4-1. Fixed disks versus floppies

Fixed Disk	Equivalent Number of Floppy Disks
10MB	29 double-sided, double-density, 360K floppies
10MB	9 quad-density, 1.2MB floppies
20MB	58 double-sided, double-density, 360K floppies
20MB	18 quad-density, 1.2MB floppies

```
A> DIR  FDISK.COM

Volume in drive A is MS330PP01
Directory of  A:\

FDISK          COM          48919   7-24-87  12:00a
            1 File(s)      5120 bytes free
```

FDISK is a menu-driven program that allows you to define how DOS can access your fixed disk. To invoke FDISK, simply type

```
A> FDISK
```

FDISK will display its main menu of options, as shown here:

```
Fixed Disk Setup Program Version 3.30
(C)Copyright Microsoft Corp. 1987

FDISK Options

Current Fixed Disk Drive: 1

Choose one of the following:

1. Create DOS partition
2. Change Active Partition
3. Delete DOS partition
4. Display Partition Information


Enter choice: [1]


Press ESC to return to DOS
```

Suppose that you want to create a DOS partition. Select menu option 1 and FDISK will display this menu:

```
Create DOS Partition
```

```
Current Fixed Disk Drive: 1

1. Create Primary DOS partition
2. Create Extended DOS partition
3. Create logical DOS drive(s) in
the Extended DOS partition


Enter choice: [1]


Press ESC to return to FDISK Options
```

Tip: *FDISK allows you to divide your fixed disk into multiple sections called partitions. Each of these partitions can contain a unique operating system.*

Some program developers place several operating systems on their disks. To do so, they divide the disk into several different sections called *partitions* (see Figure 4-6). However, most users will want to use only one operating system—

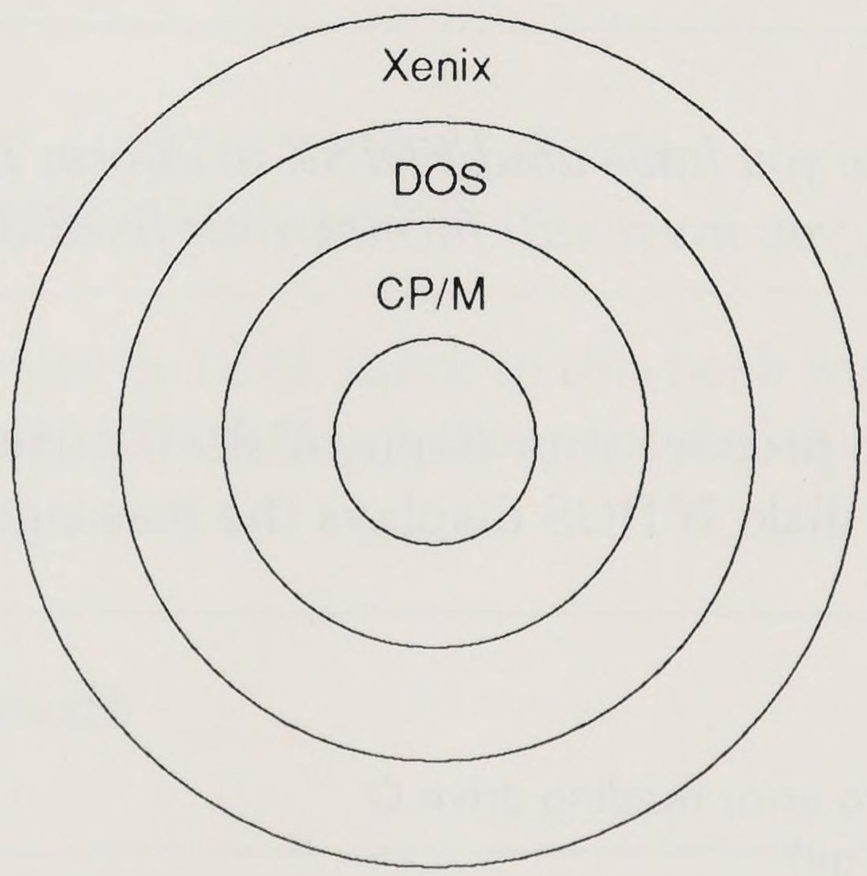


Figure 4-6. Partitions on fixed disk

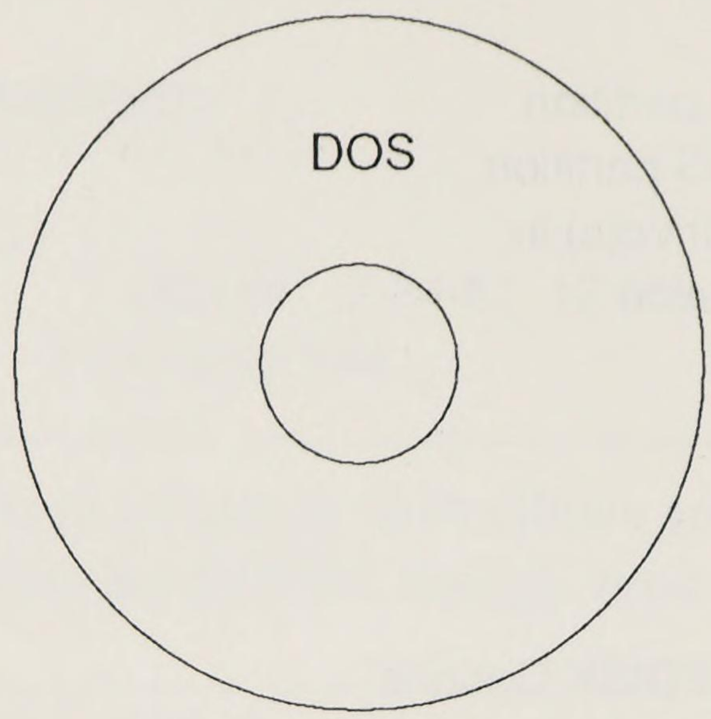


Figure 4-7. Defining a partition

DOS—and will define only one area (or partition) on their disk by using FDISK (see Figure 4-7).

In many cases, you computer retailer will have performed this step for you. When you have used the FDISK command, accessing your fixed disk results in

```
A> DIR C:

General Failure error reading drive C
Abort, Retry, Fail?
```

Trap: *Once you have used FDISK to inform DOS that your fixed disk exists, you must still format your fixed disk to make it usable.*

Just as you had to prepare your floppy disks for use under DOS, you must also format your fixed disk. If DOS displays the message

```
A> DIR C:

General Failure error reading drive C
Abort, Retry, Fail?
```

you must place your DOS Program Disk in drive A and issue the command

```
A> FORMAT C: /S
```

Since you want DOS to boot from your fixed disk, you must include the /S qualifier. When the FORMAT command completes, a directory listing of drive C reveals

```
A> DIR C:

Volume in drive C has no label
Directory of C:\

COMMAND      COM          25276   7-24-87  12:00a
            1 File(s)     20110848 bytes free
```

Using the DOS LABEL command, assign a name to your fixed disk, as shown here:

```
A> LABEL C:DOSDISK
```

A directory listing of the disk now displays

```
A> DIR C:

Volume in drive C is DOSDISK
Directory of C:\

COMMAND      COM          25276   7-24-87  12:00a
            1 File(s)     20110848 bytes free
```

Your fixed disk is ready for use by DOS. Later in this book we will examine how to correctly copy the remaining DOS files to your fixed disk.

Examining the Current State of Your Disks

As stated earlier in this chapter, your disks may have damaged locations (sectors) where DOS is unable to store information. The FORMAT command marks

those locations as unusable for DOS. It is also possible, although not common, for DOS to encounter errors when it writes information to your disk. Such errors can leave your disks in a partially completed state. The DOS CHKDSK command allows you to examine the current state of your disk.

Tip: *The DOS CHKDSK command examines the current state of your disks.*

CHKDSK displays the following:

- The disk volume label and the date that the volume label was assigned
- The number of bytes of data that the disk can store
- The number of hidden files on your disk and the amount of disk space (in bytes) that these files consume
- The amount of space consumed by user files on the disk (in bytes)
- The number of bytes available on the disk for storing information
- The amount of memory in your computer along with the amount of memory currently unused by DOS
- An error message if damaged files exist
- All of the files on your disk (optional)
- Information about disk contiguity (optional)

CHKDSK can also recover damaged files for your examination.

Invoking CHKDSK at the DOS prompt results in output similar to this:

```
A> CHKDSK  B:

362496 bytes total disk space
9216 bytes in 7 user files
353280 bytes available on disk

655360 bytes total memory
584544 bytes free
```

In this case, the disk that CHKDSK examined did not contain any damaged locations. Therefore, CHKDSK displayed only its standard information.

Notice the message about a hidden file. When you create a bootable disk (using FORMAT /S), DOS must copy several files to the disk to allow your computer to start DOS. Because these files are critical to your computer operations, DOS hides them from your directory listing (making them *hidden files*). This prevents you from inadvertently deleting them. The only way to verify that these files exist is to use CHKDSK. Also, if you assign a volume name to your disk by using the DOS LABEL command, DOS places the volume name in a hidden file.

Using the CHKDSK command on your DOS Program Disk, for example, reveals

```
A> CHKDSK

Volume MS330PP01   created Jul 24, 1987 11:28a

362496 bytes total disk space
53248 bytes in 3 hidden files
304128 bytes in 34 user files
5120 bytes available on disk

655360 bytes total memory
584544 bytes free
```

Notice the memory values displayed at the bottom of the output. The first value tells you how much memory you have present in your computer, and the second value tells you how much of that memory is currently unused by DOS.

655360 bytes total memory	←	Memory in computer (640K)
584544 bytes free	←	Available memory

Tip: *To display the names of all the files on your disk, use the CHKDSK /V command.*

CHKDSK will optionally display the names of all files on your disk. To select this option, use the /V qualifier, as shown here:

```
A> CHKDSK /V
```

In most cases, you will simply use the DIR command to list the files on your disk. However, as you will learn later in this book, there are times when DIR can only list a portion of the files. The CHKDSK /V option always allows you to display all of the files on your disk.

If CHKDSK locates damaged files on your disk, it will display a message in the form

```
nnn lost clusters found in n chains.
Convert lost chains to files (Y/N)?
```

Although CHKDSK asks whether you want the files recovered, regardless of how you respond to the prompt, it will not repair them.

Trap: *By default, CHKDSK will not correct damaged locations on your disk. To direct CHKDSK to repair such locations, use the /F qualifier.*

If CHKDSK encounters partial files on your disk and displays the message

```
nnn lost clusters found in n chains.
Convert lost chains to files (Y/N)?
```

simply reinvoke CHKDSK with the /F (fix) qualifier, as shown here:

```
A> CHKDSK B:/F
```

In this case, when you respond to the prompt

```
nnn lost clusters found in n chains.
Convert lost chains to files (Y/N)?
```

with **Y**, CHKDSK will recover the damaged files, assigning them names in the form FILE0001.CHK. By using the DOS TYPE command, you can examine the contents of these files on your screen display:

```
A> TYPE FILE0001.CHK
```

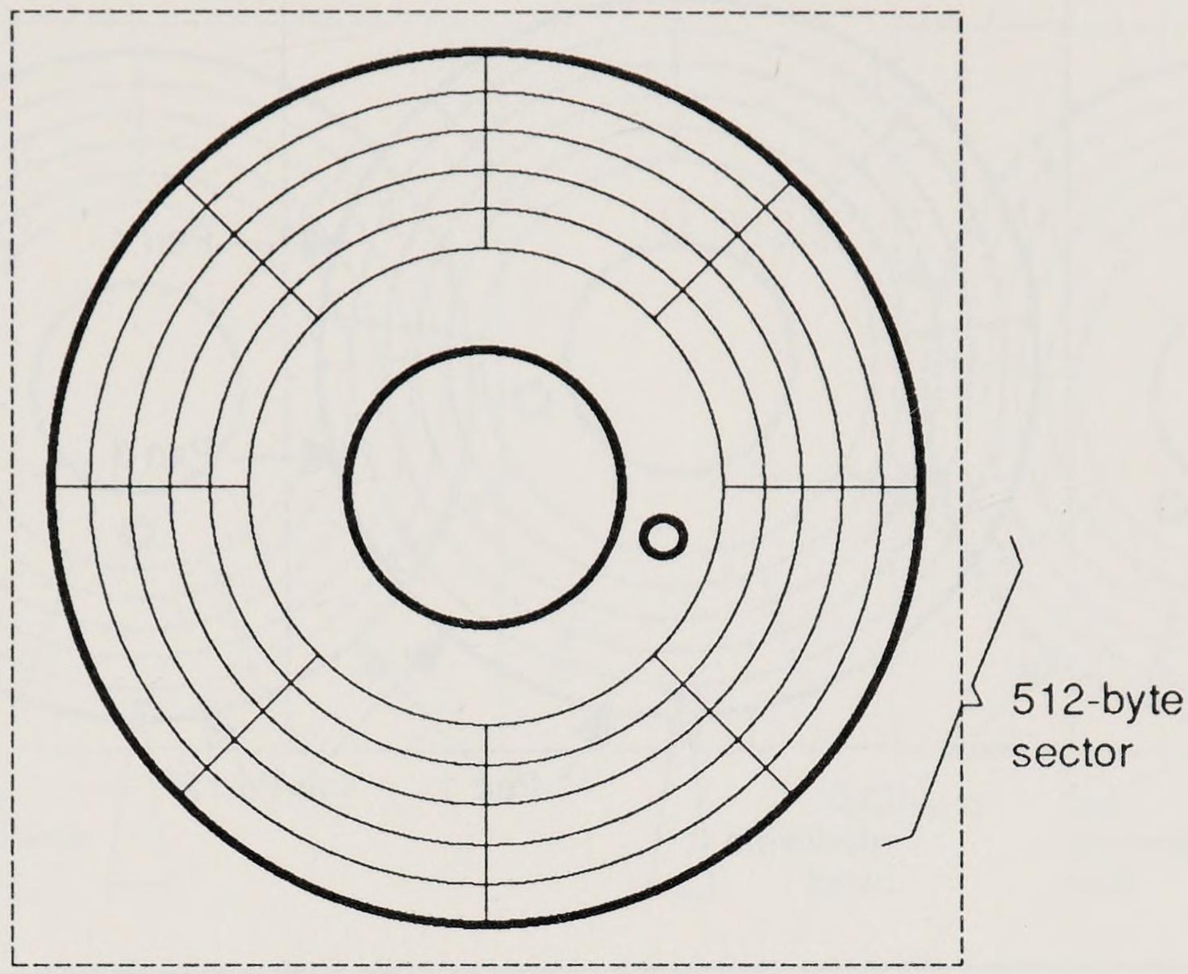


Figure 4-8. Segment on a floppy disk

If the file contains useful information, rename it appropriately. If the file contains unrecognizable characters, delete the file, freeing disk space.

Contiguous Files

Remember that DOS divides your disk into circular rings called cylinders. Within each cylinder, DOS further divides your disk into smaller pieces called *segments.* Each of these segments, or *sectors,* can store 512 bytes (characters) of information (see Figure 4-8).

When DOS reads or writes information to your disk, it does so by rapidly spinning your disk around in the drive in order to locate the desired segment. When your disks are new, DOS can store your files in consecutive segments, as shown in Figure 4-9.

The advantage of having your files stored in contiguous locations is speed of access. Once DOS reads the first part of the file, it needs only to rotate the disk a small distance in order to read the second part (see Figure 4-10).

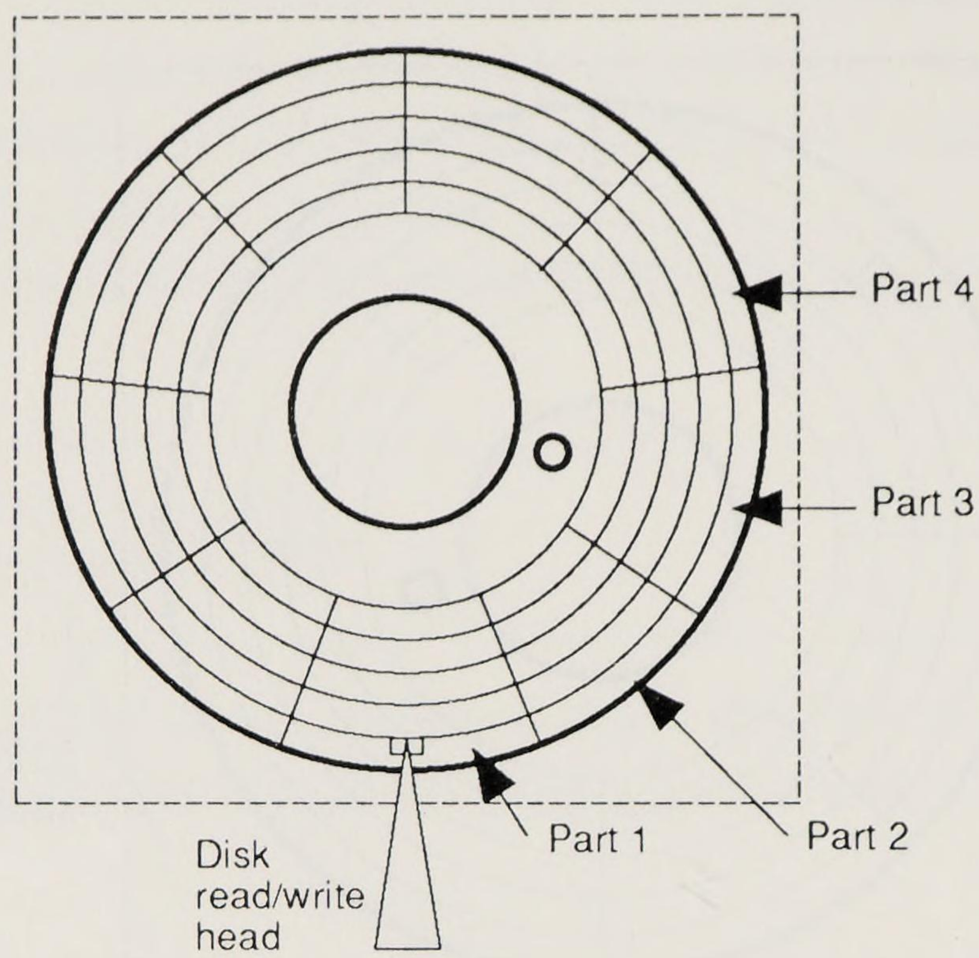


Figure 4-9. Consecutive sectors

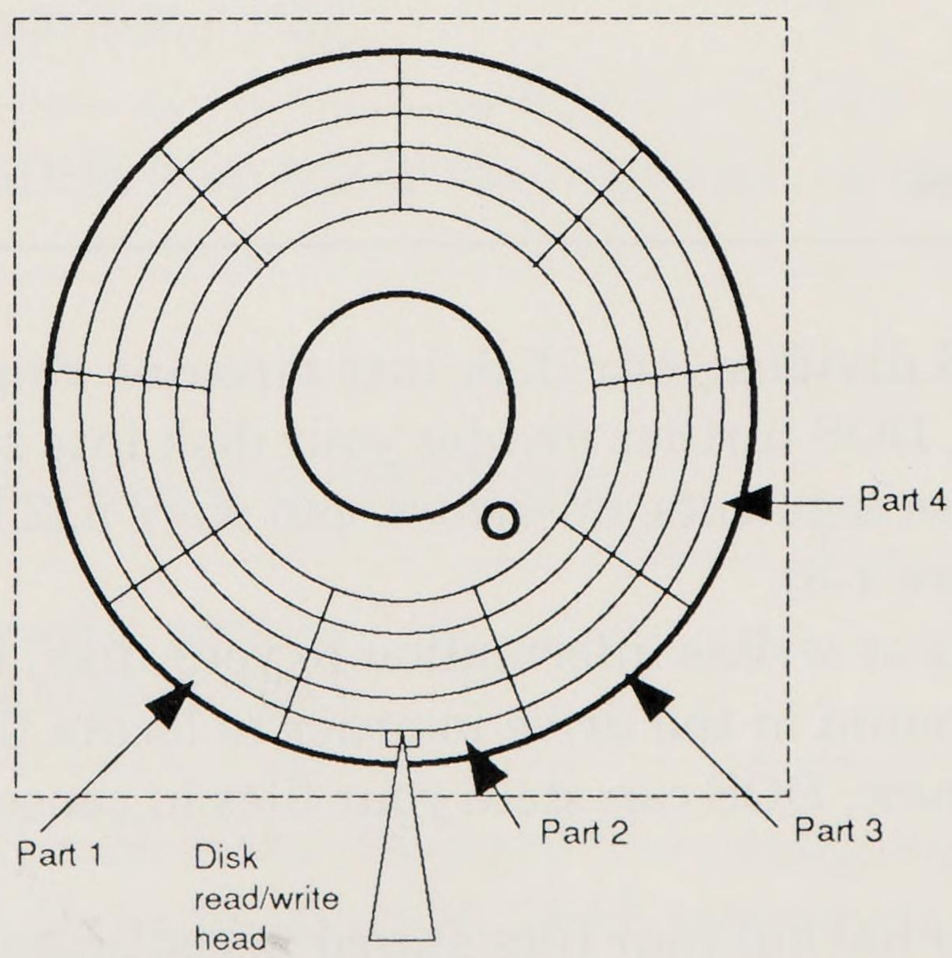


Figure 4-10. DOS reading the second part of a file

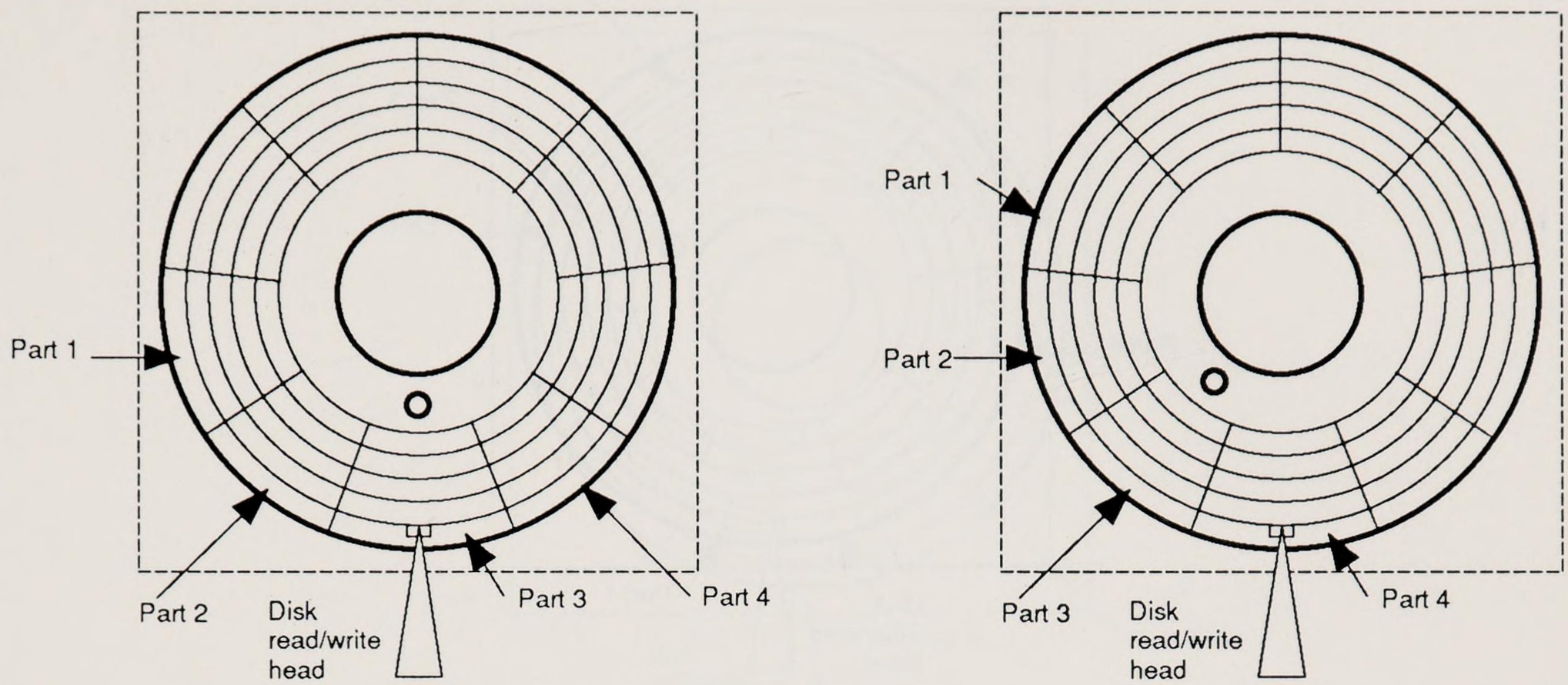


Figure 4-11. DOS reading the third and fourth parts of a file

This process repeats for the third and fourth parts of the file (see Figure 4-11).

As you use your disks on a regular basis, DOS is no longer afforded the luxury of keeping your files in adjoining locations on disk. This is because, as you create and delete files, you make sectors available for storing information about the disk. When it comes time for DOS to store your file, it must place the file on disk anywhere it will fit. As a result, portions of your file may end up widely dispersed about your disk (see Figure 4-12).

Once DOS reads the first part of your file, it must rotate the disk half a revolution in order to read the second part, as shown in Figure 4-13. Next, to read the third part of the file, your disk must make an almost complete revolution, as shown in Figure 4-14. To read the fourth part of the file requires one more partial rotation.

As the size of your files increases, these additional disk rotations can become quite time-consuming.

Tip: *The CHKDSK command allows you to determine whether a file or disk is contiguous.*

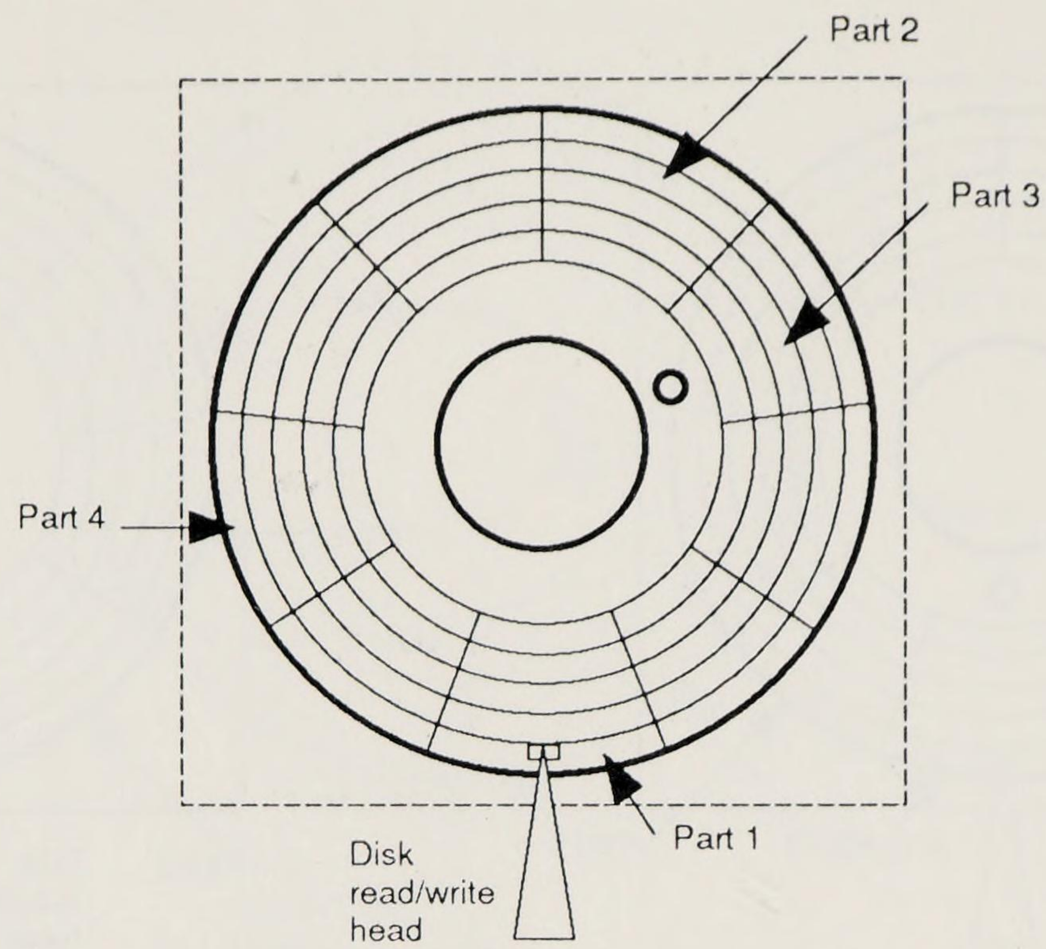


Figure 4-12. Widely dispersed sections of a file

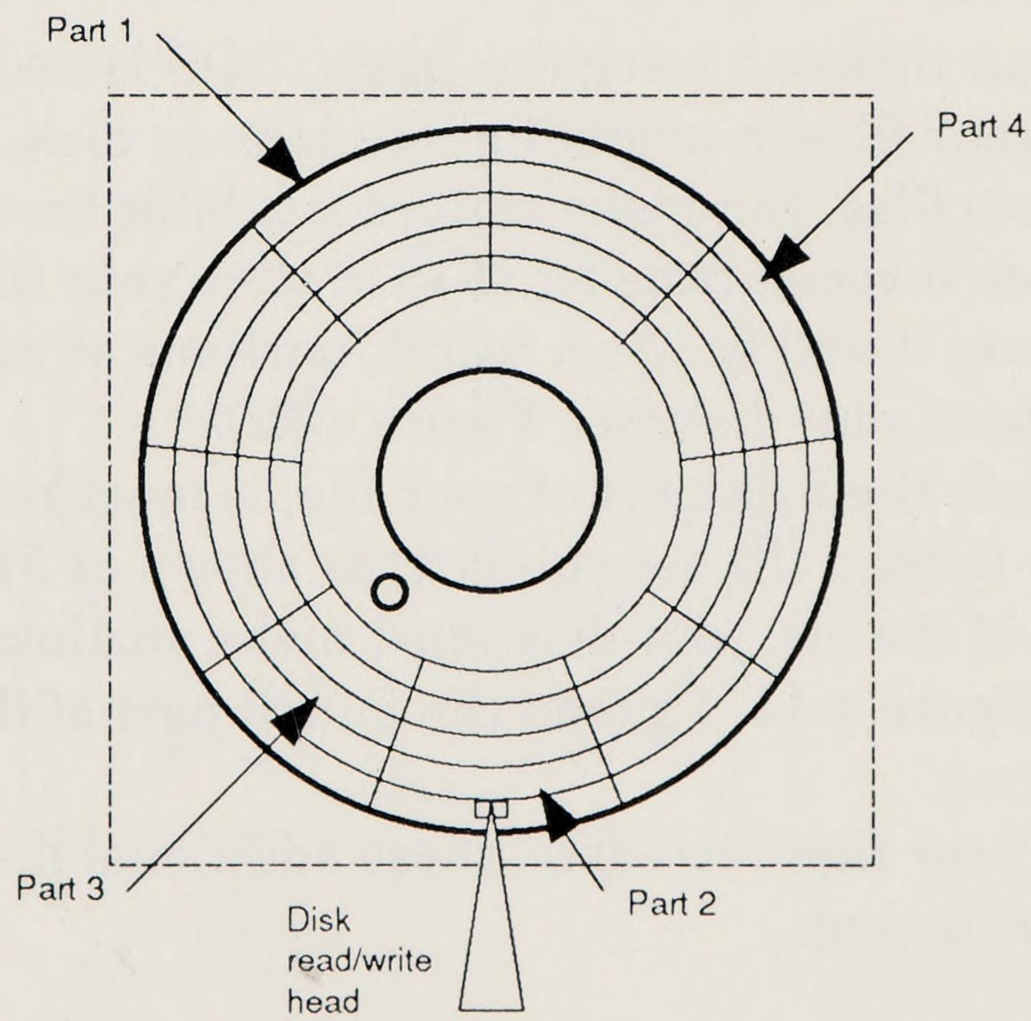


Figure 4-13. DOS rotating a disk half a revolution

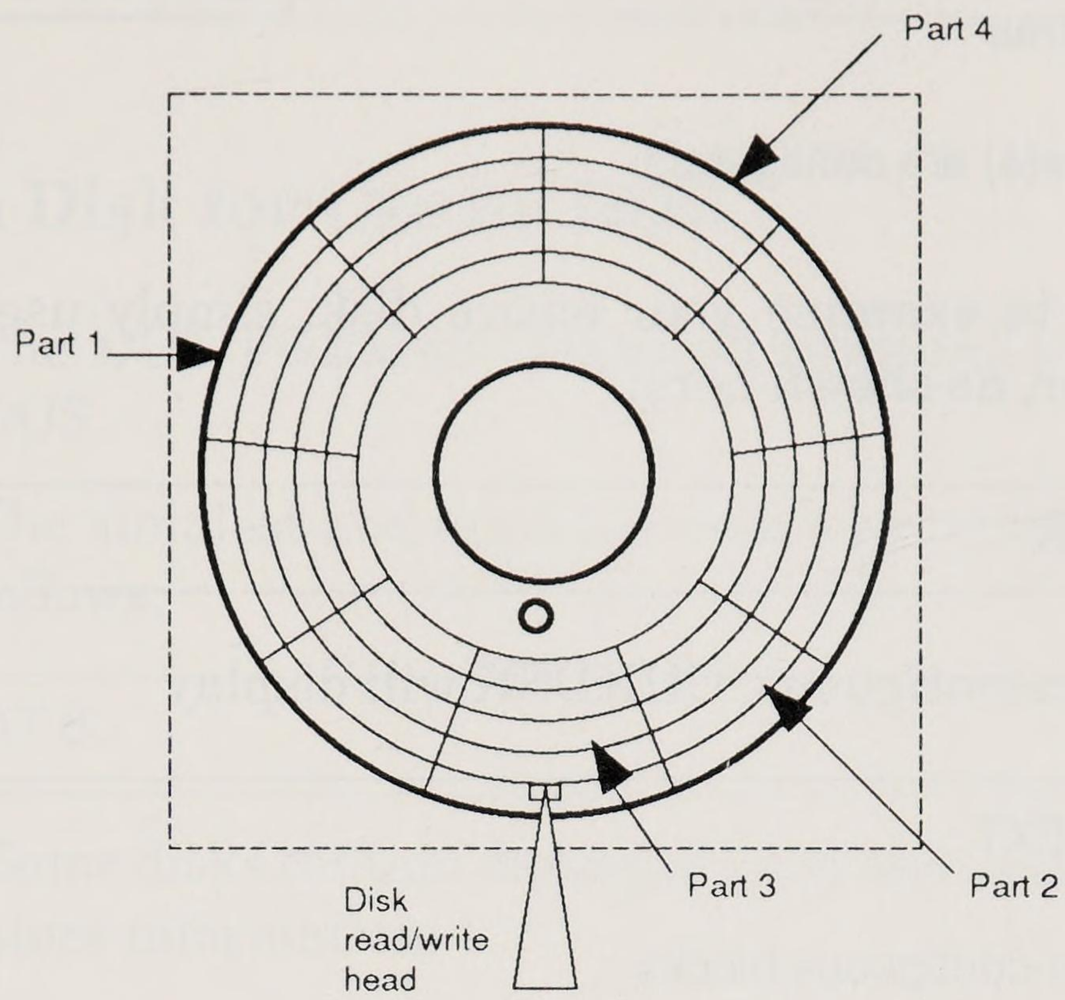


Figure 4-14. DOS rotating a disk a full revolution

To examine a specific file for disk contiguity, simply type the file's name in the CHKDSK command line, as shown here:

```
A> CHKDSK DISKCOPY.COM
```

CHKDSK will display its standard information and then report on the file, as shown here:

```
A> CHKDSK DISKCOPY.COM

Volume MS330PP01  created Jul 24, 1987 11:28a

362496 bytes total disk space
53248 bytes in 3 hidden files
304128 bytes in 34 user files
5120 bytes available on disk
```

```
655360 bytes total memory
584544 bytes free

All specified file(s) are contiguous.
```

If you want to examine your entire disk, simply use the DOS asterisk wildcard character, as shown here:

```
A> CHKDSK  B:*.*
```

If a file is noncontiguous, CHKDSK will display

```
B:FILENAME.EXT

Contains n non-contiguous blocks.
```

If your disk contains many files that are not contiguous, the only way to correct the problem is with a complete file backup and restore operation, which will be discussed later in this book.

Most users do not worry about whether their files are contiguous. However, if you are trying to maximize your computer's potential, CHKDSK gives you a tool for examining your disk.

Tips and Traps Summary

Preparing a Disk for Use by DOS

Tip: The DOS FORMAT command prepares a disk for use by DOS.

Tip: The simplest and most common way to use FORMAT is as follows:

```
A> FORMAT B:
```

Trap: Some disks contain damaged locations (sectors) that cannot store information.

Trap: Once you format a disk, you destroy any information previously stored on it.

Tip: FORMAT also allows single-disk drive systems to prepare disks for use by DOS.

Trap: Just because a disk has been formatted for use under DOS does not mean that the disk can start (boot) DOS.

Tip: To create a disk that can start DOS, you must use the /S qualifier in the FORMAT command.

Trap: Not all versions of DOS copy the COMMAND.COM file to the target disk during formatting.

Tip: Do not format all of your disks as bootable.

Trap: Not all DOS floppy disks are readily interchangeable.

Tip: The FORMAT command with the /4 qualifier allows you to format a 360K disk in your 1.2MB disk drive.

Naming Your Disks

Tip: Assigning unique names to your disks improves your disk organization.

Tips and Traps Summary (*continued*)

Tip: The DOS LABEL command allows you to assign an 11-character name to your disks.

Trap: If the volume name that you enter is invalid, LABEL will display an error message and prompt you to enter a new label.

Tip: LABEL allows you to specify the disk label in the LABEL command line.

Tip: To delete a disk volume label, simply press ENTER when LABEL prompts you for a volume name.

Tip: The VOL command displays the disk volume label for the disk drive you specify.

Preparing a Fixed Disk for Use by DOS

Trap: If DOS has never been told that your computer has a fixed disk, it will display an error message each time you attempt to access the disk.

Tip: FDISK allows you to divide your fixed disk into multiple sections called partitions. Each of these partitions can contain a unique operating system.

Trap: Once you have used FDISK to inform DOS that your fixed disk exists, you must still format your fixed disk to make it usable.

Examining the Current State of Your Disks

Tip: The DOS CHKDSK command examines the current state of your disks.

Tip: To display the names of all the files on your disk, use the CHKDSK /V command.

Tips and Traps Summary (*continued*)

Contiguous Files

Tip: By default, CHKDSK will not correct damaged locations on your disk. To direct CHKDSK to repair such locations, use the /F qualifier.

Tip: The CHKDSK command allows you to determine whether a file or disk is contiguous.

5

Using System Devices

As stated in Chapter 1, your computer is comprised of several standard hardware devices including the screen, keyboard, and chassis (see Figure 5-1). Many computers have devices attached such as a printer, mouse, and modem, as shown in Figure 5-2. DOS assigns unique names to these devices, and in this chapter you will learn how to access them by using their device names.

Creating Your Own Text Files

Chapter 3 examined DOS files in detail. Remember that you can assign a specific 3-character extension to group files by function. You also learned that DOS provides several files with the following extensions.

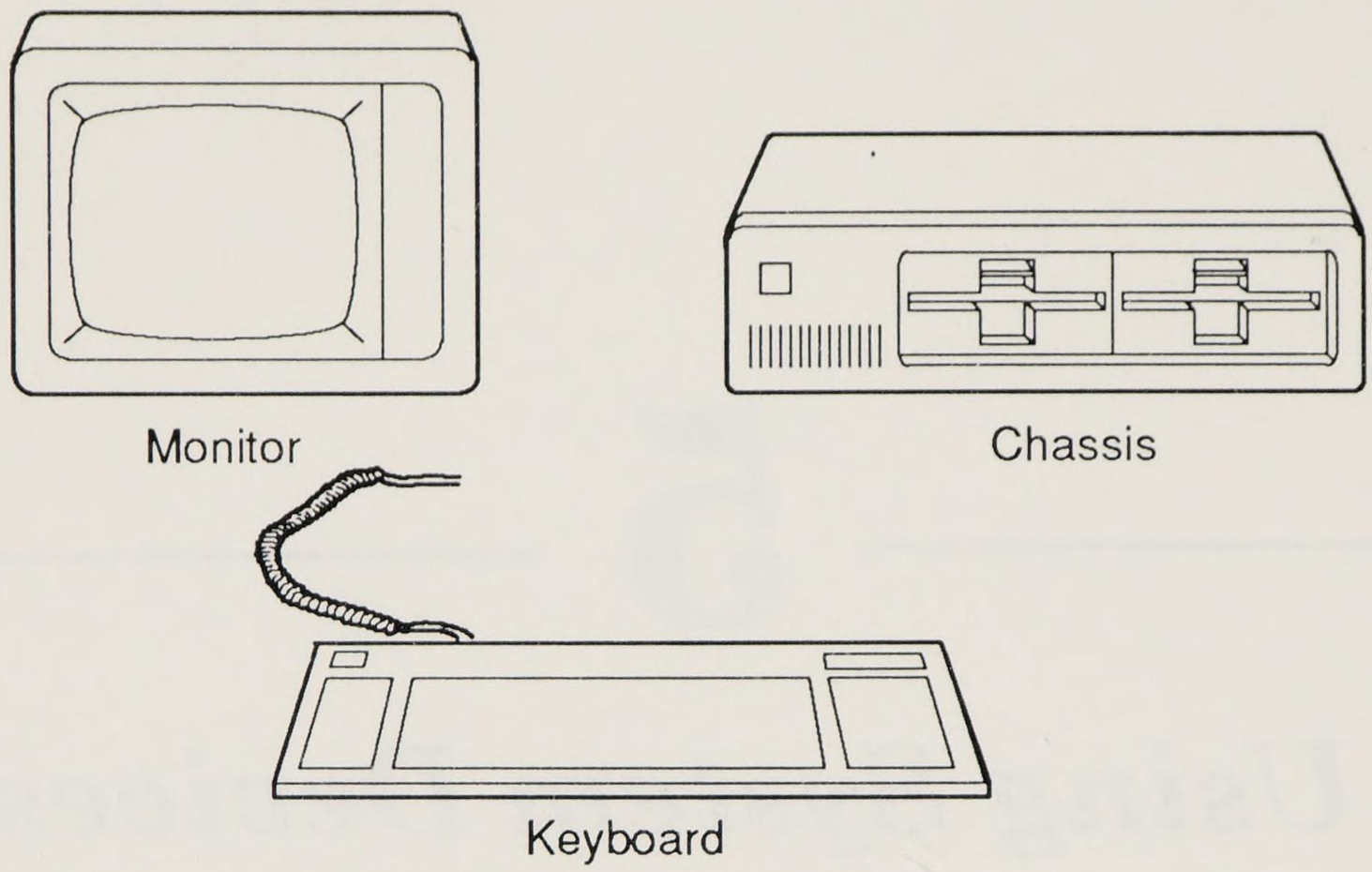


Figure 5-1. Standard hardware devices

EXE	Executable DOS program
COM	Executable DOS command
SYS	Operating system file
CPI	Code page information file

By examining a file's extension, you can quickly determine its contents.

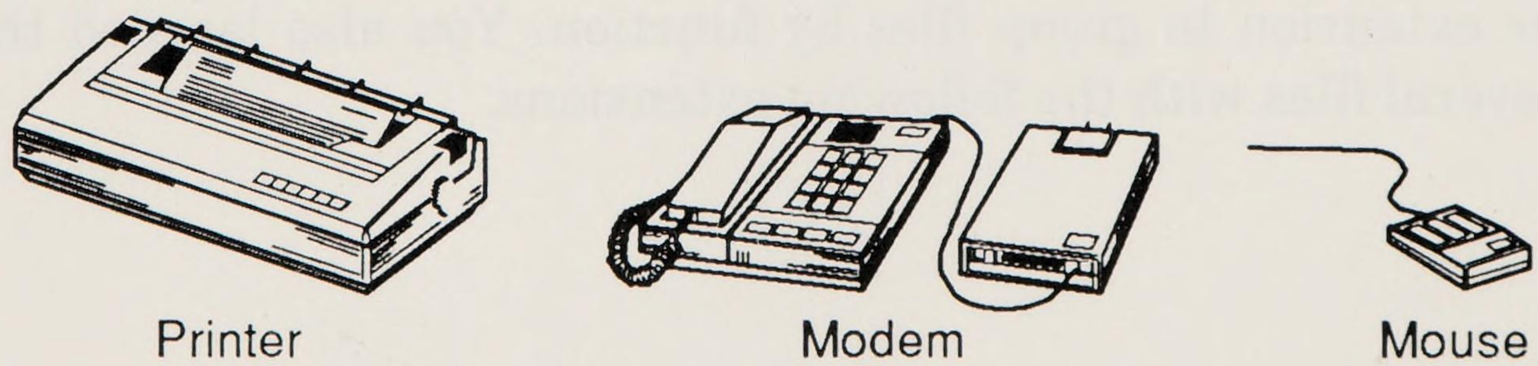


Figure 5-2. Additional devices added to computer system

Files exist to store information and programs. In this section, you will learn one method of creating files that contain your own information. Unlike files containing DOS programs, the files that you create will contain standard letters and numbers. These files are called *text files.*

Tip: *By using the device name CON as your source of information, you can create files with the COPY command.*

In Chapter 3 you learned that the command

```
A> COPY COMMAND.COM B:FILENAME.EXT
```

directs DOS to copy the contents of the COMMAND.COM file from the current disk to the disk in drive B, assigning the new file the name FILENAME.EXT. COMMAND.COM is the source for the file copy, while B:FILENAME.EXT is the target, as shown here:

```
A> COPY COMMAND.COM B:FILENAME.EXT
       \_________/  \_________/
        Source       Target
```

In a similar manner, DOS allows you to use the device name CON to reference your keyboard when you create a file. Given the command

```
A> COPY CON B:FILENAME.EXT
```

DOS will copy all the information that you type at your keyboard into the B:FILENAME.EXT file. In this case, your keyboard is the source of the information to the COPY command.

```
A> COPY CON B:FILENAME.EXT
       \__/  \_________/
       Source   Target
```

Assuming that your system has two floppy-disk drives, enter the following command.

```
A> COPY CON B:TEST.DAT
```

If you have only one floppy-disk drive, remove the disk drive specifier from the target file throughout the remaining examples, yielding

```
A> COPY CON TEST.DAT
```

In either case, when you press ENTER, DOS will move your cursor down one line, as shown here:

```
A> COPY CON B:TEST.DAT
_
```

DOS is waiting for you to enter the first line of the file. Type the line

```
This is the first line
```

When you press ENTER, DOS again moves the cursor down one line:

```
A> COPY CON B:TEST.DAT
This is the first line
_
```

Now type

```
The second line is here
```

Repeat this process for the third line:

```
A> COPY CON B:TEST.DAT
This is the first line
The second line is here
Line three is the last time
_
```

Assuming that you want to enter only three lines, you must inform DOS that you are done with the file. To do so, press the key labeled F6 (see Figure 5-3). DOS will respond by displaying

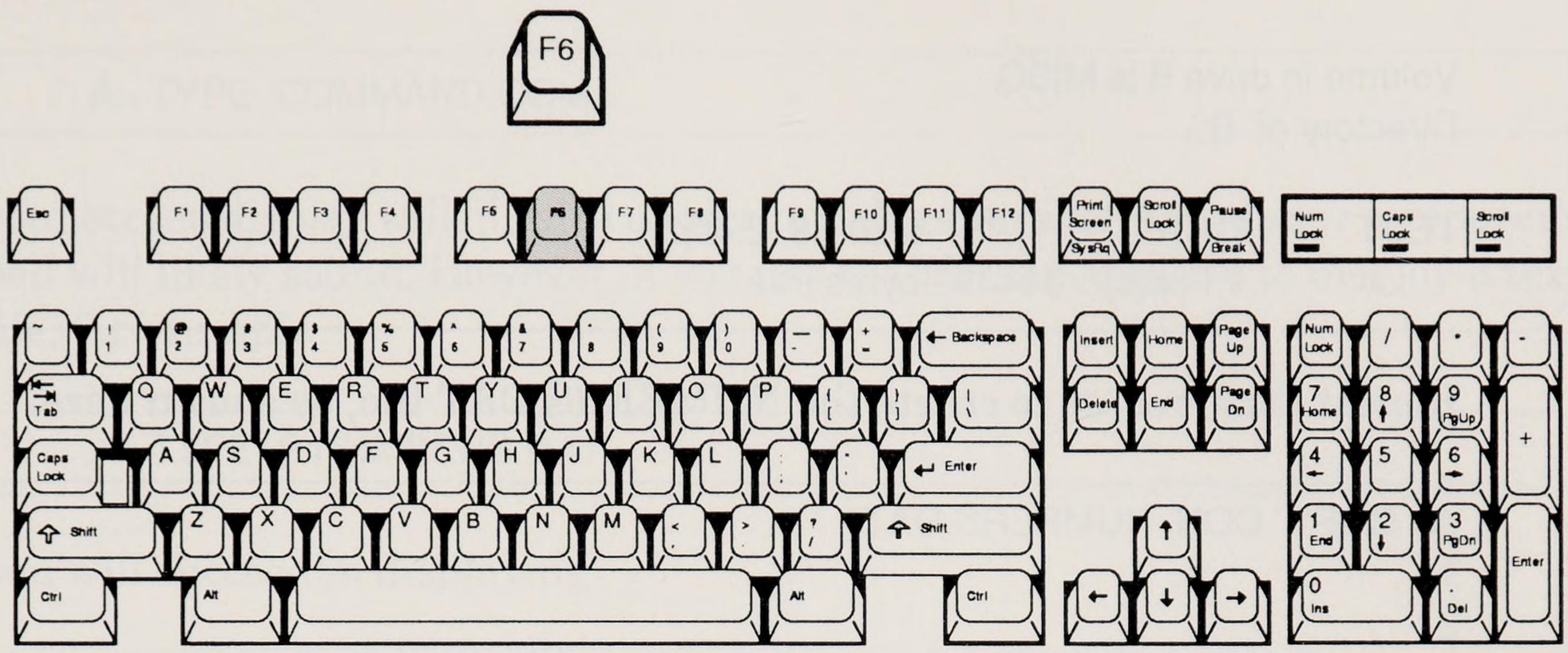


Figure 5-3. Location of F6 key

```
A> COPY  CON  B:TEST.DAT
This is the first line
The second line is here
Line three is the last time
^Z
```

^Z is how DOS represents the end of a file. Since you are at the end of your file, you must press F6 to inform DOS to place the end-of-file marker in your file. Press ENTER and DOS will create your file, as shown here:

```
A> COPY  CON  B:TEST.DAT
This is the first line
The second line is here
Line three is the last time
^Z
            1 File(s) copied
```

A directory listing of TEST.DAT reveals

```
A> DIR  B:TEST.DAT

Volume in drive B is MISC
Directory of  B:\

TEST        DAT          78   6-20-88  2:40p
            1 File(s)   304496 bytes free
```

Repeat this process to create the NUMBERS.DAT file, as shown here:

```
A> COPY  CON  NUMBERS.DAT
1
2
3
4
5
^Z
            1 File(s) copied


A>
```

Using the DOS TYPE command, you can display the contents of these files.

Tip: *The DOS TYPE command displays the contents of a text file.*

As stated in Chapter 3, files contain programs and other information. A text file contains standard letters and numbers that would normally appear in a letter, book, or other "text." Files with the extension COM or EXE are not text files. Instead, they contain nonprintable characters that DOS cannot display on your screen.

Trap: *The DOS TYPE command cannot display all files. Files with the extension COM or EXE contain nonprintable characters that DOS cannot display on your screen.*

For example, if you issue the command

```
A> TYPE COMMAND.COM
```

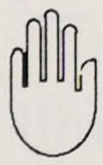
your screen display will fill with unrecognizable characters, and your computer's bell will likely sound. However, if you use the TYPE command to display a text file, for example

```
A> TYPE B:NUMBERS.DAT
```

you will succeed in displaying

```
A> TYPE B:NUMBERS.DAT
1
2
3
4
5

A>
```

TYPE is an internal DOS command that always resides in your computer's memory. If TYPE displays an error message, it is normally because DOS cannot locate the file to display.



Trap: *If TYPE displays the error message*

```
File not found
```

it could not locate the file you specified for display.

If this error message occurs, you have probably misspelled the file name to display. If this is not the case, use the DOS DIR command to locate the file.

Tip: *In order to cancel a COPY CON FILENAME.EXT command, use the CTRL-C key combination.*

If, after issuing the command

```
A> COPY CON FILENAME.EXT
```

you decide that you don't want to create the file you specified, hold down the key labeled CTRL and press the C key. DOS will terminate your command without creating the file, as shown here:

```
A> COPY CON FILENAME.EXT
A
B
C
D
^C

A>
```

Trap: *COPY overwrites any existing file of the same name as your target file.*

If you issue the command

```
A> COPY CON FILENAME.EXT
```

and a file with the name FILENAME.EXT already exists on disk, COPY will overwrite the existing file's contents. Therefore, use the DIR command to verify that no files exist on disk with names that match your chosen file name.

Tip: *Use meaningful file names.*

As stated in Chapter 3, make your DOS file names as meaningful as possible. By examining a file's name, you should have a good idea of the file's contents.

Using CON as an Output Device

Just as you can use the CON device for the source of your file copy operations, you can also use it for your target, as shown here:

```
A> COPY B:NUMBERS.DAT CON
```

Tip: *For input operations, CON points to your keyboard. For output operations, CON points to your screen display.*

If the device name CON appears in a COPY command, ask yourself the following question: "Is CON the source or target for the file copy?"

If CON is the source of the information to copy, it references your keyboard, as shown in Figure 5-4. If CON is the target of your file copy, it references your screen display, as shown in Figure 5-5.

You will normally use the TYPE command in place of

```
A> COPY FILENAME.EXT CON
```

However, you should know that this capability exists.

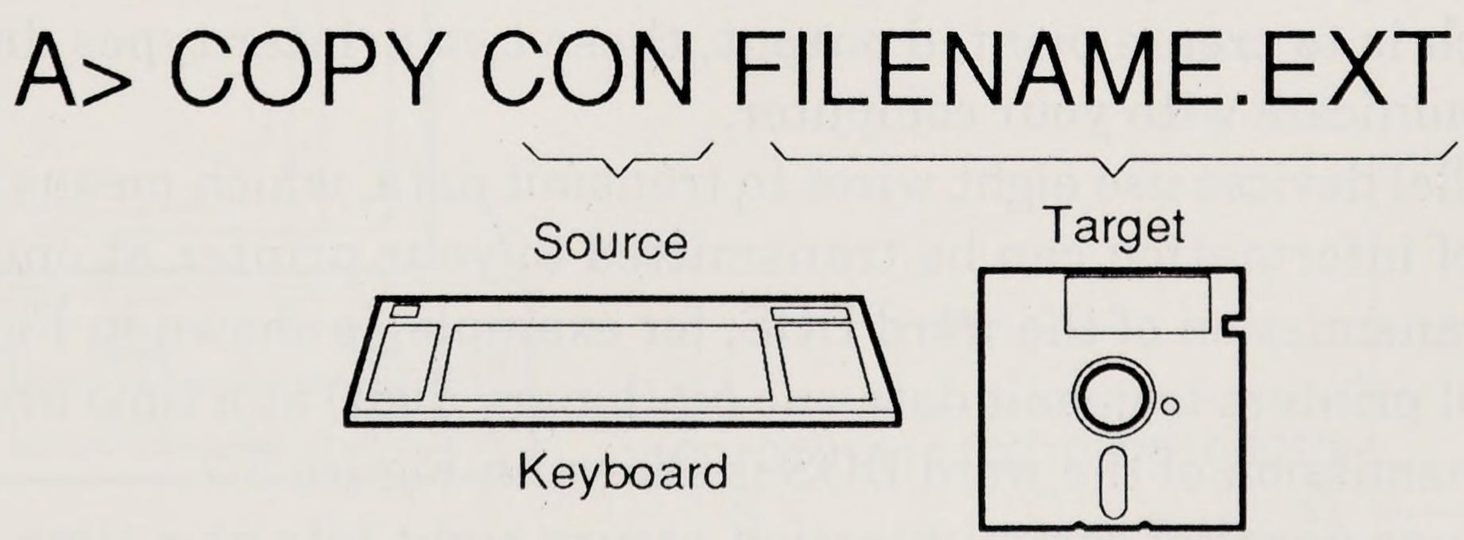


Figure 5-4. CON referencing the keyboard

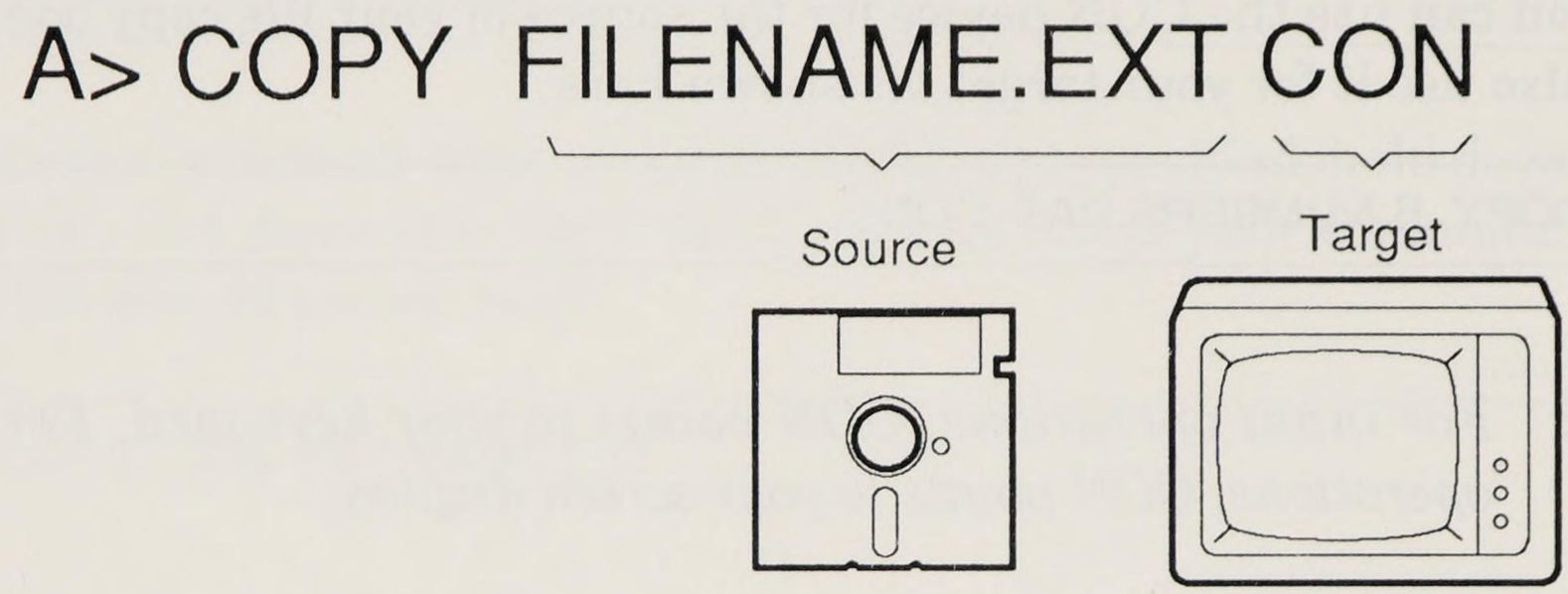


Figure 5-5. CON referencing the screen display

Looking at DOS Device Names

Most of you have a printer attached to your computer. Depending on the type of printer you are using, the manner in which you reference the printer and the preparatory steps you must perform in order to access the printer will differ.

The two printer types available are *parallel* and *serial*. Although the function of each is to create printed output, these two printer types differ in how they communicate with your computer.

Parallel devices use eight wires to transmit data, which means that an entire byte of information can be transmitted to your printer at one time. The parallel transmission of the word DOS, for example, is shown in Figure 5-6.

Serial printers transmit data one *bit* (binary digit) at a time over one wire. Serial transmission of the word DOS is shown in Figure 5-7.

Because parallel communication occurs eight bits at a time, it is much faster than serial communication. Most of the printers in use today are parallel printers.

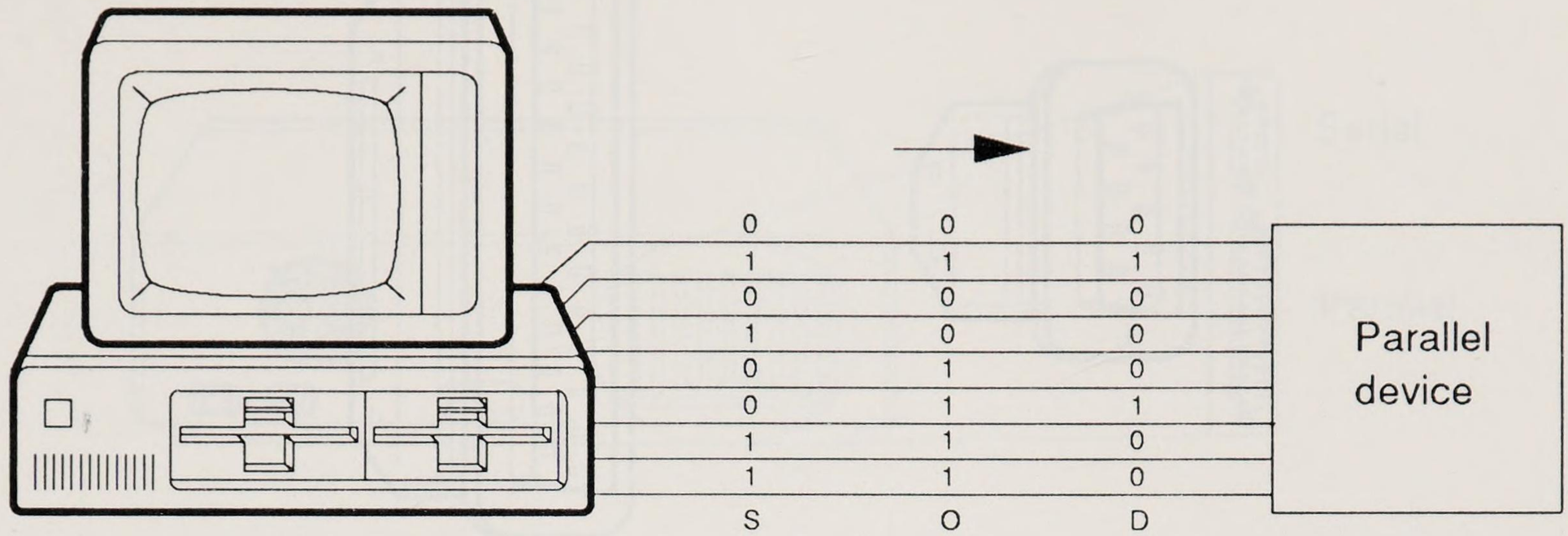


Figure 5-6. Parallel data communication

Tip: *Determine the type of printer that you are using, serial or parallel. The steps you must perform to use your computer are dependent upon the printer type.*

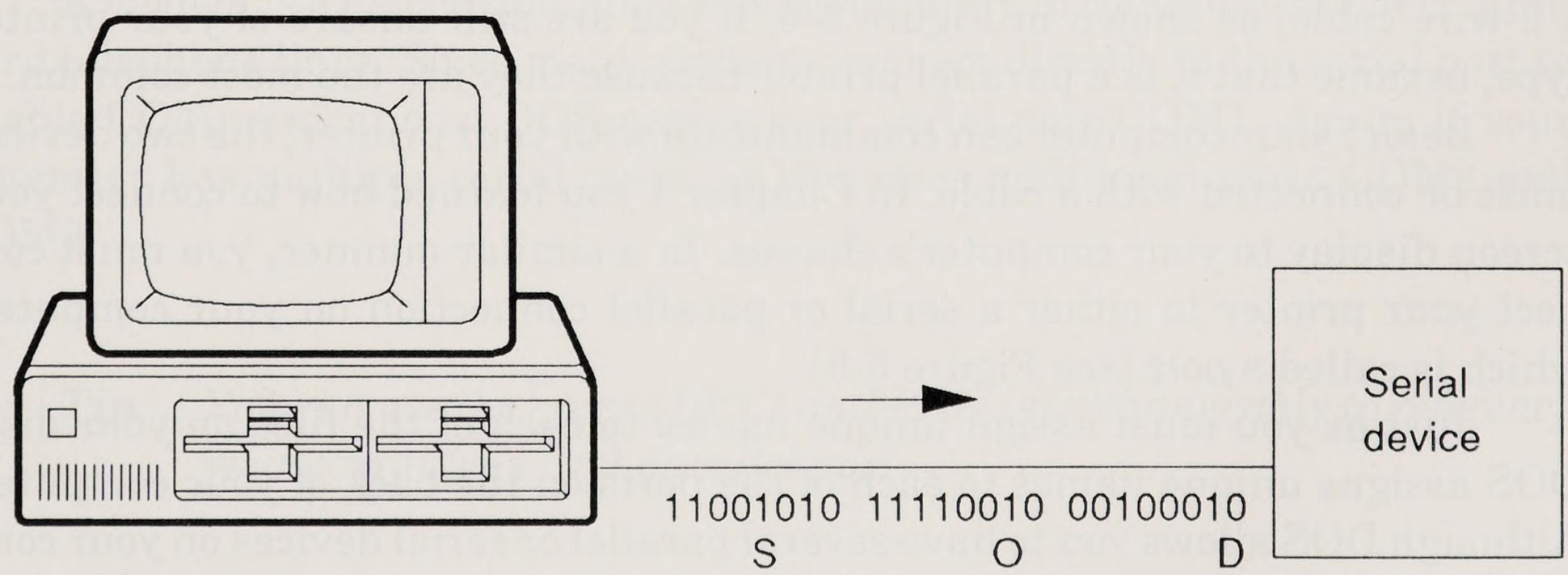


Figure 5-7. Serial data communication

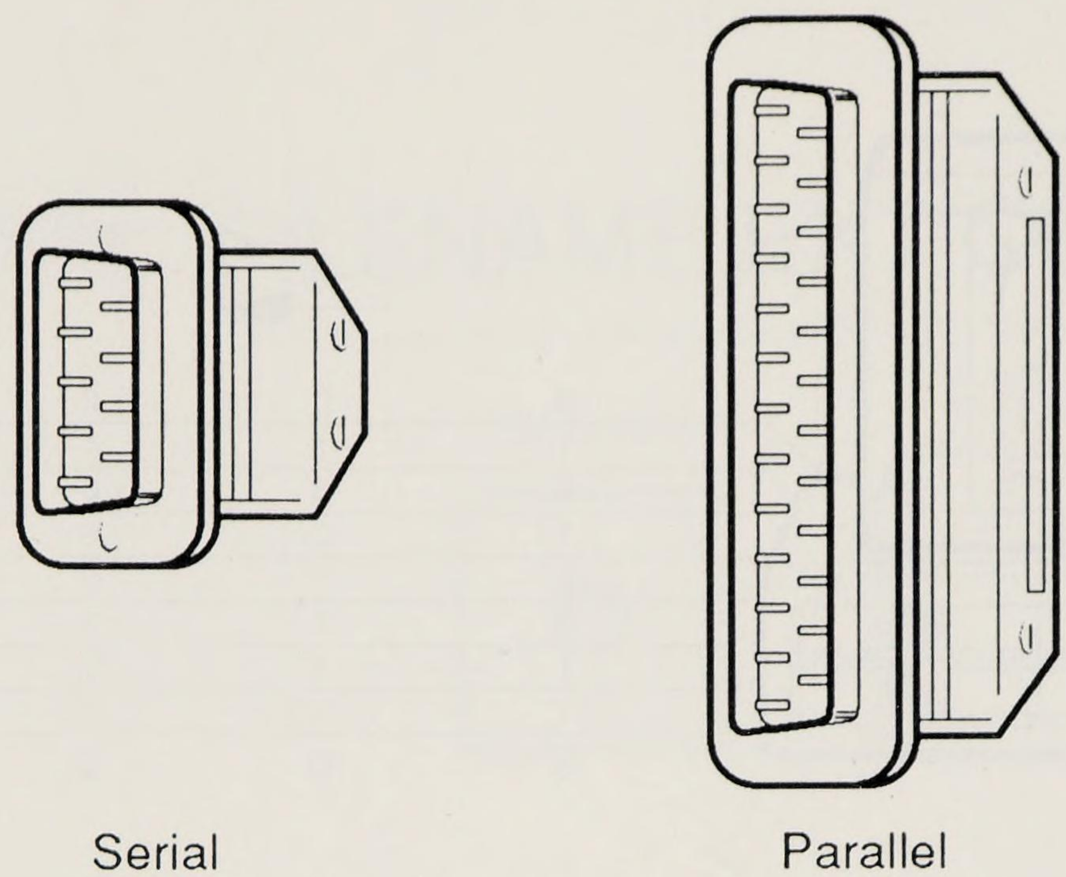


Figure 5-8. Parallel and serial connectors

If your printer manual is readily available, read it to determine your printer type. If you do not have a manual, examine the cables that connect your printer to your computer. Parallel printers use a 25-wire cable, while serial printers use a 9-wire cable, as shown in Figure 5-8. If you are still unsure of your printer type, assume that it is a parallel printer because they are the most common.

Before your computer can communicate with your printer, the two devices must be connected with a cable. In Chapter 1 you learned how to connect your screen display to your computer's chassis. In a similar manner, you must connect your printer to either a serial or parallel connection on your computer, which is called a *port* (see Figure 5-9).

Just as you must assign unique names to each of the files on your disk, DOS assigns unique names to each of the ports on the back of your computer. Although DOS allows you to have several parallel or serial devices on your computer, most users have only one.

Tip: *DOS names your parallel port LPT1 and your serial port COM1.*

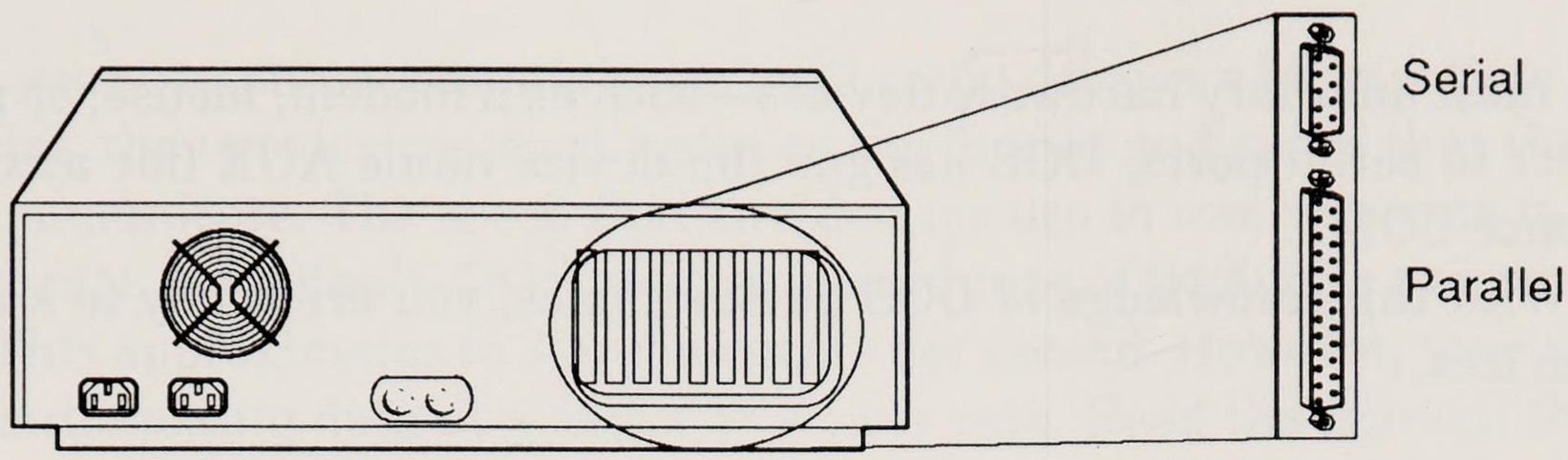


Figure 5-9. Serial and parallel ports

In the same way that DOS assigns the device name CON to your screen display and keyboard, it also names your parallel and serial ports. Your parallel port is named LPT1 because most users connect their Line PrinTers (LPT) to this port. If you have multiple parallel ports on your system, DOS names additional ports LPT2 and LPT3.

A *modem* is a device that allows two computers to communicate over standard telephone lines. Since many modems connect directly to the serial port to enable COMmunications, DOS names your serial port COM1. Again, if your computer has multiple serial ports, DOS names additional ports COM2 and COM3.

Tip: *You can use the names LPT1 and PRN interchangeably to reference your computer's first parallel port.*

Because parallel printers are the most common printers used with personal computers and since most of these printers are connected to LPT1, DOS assigns the name PRN (for printer) to LPT1.

Tip: *You can use the names COM1 and AUX interchangeably to reference your computer's first serial port.*

Since most auxiliary hardware devices—such as a modem, mouse, or printer—connect to serial ports, DOS assigns the device name AUX (for auxiliary) to reference COM1.

With this knowledge of DOS device names, you are ready to start using the devices.

Using Your Printer

As just discussed, a parallel printer will normally be connected to LPT1 (or PRN) while a serial printer will be connected to COM1 (or AUX). With this in mind, you can print the files that you created earlier in this chapter by issuing either the command

```
A> COPY B:TEST.DAT LPT1
```

or the command

```
A> COPY B:TEST.DAT COM1
```

Turn on your printer and make sure that the on-line light is illuminated. Then, issue one of the commands just listed.

Trap: *If DOS cannot access your printer, it will display the message*

```
Not ready error writing device LPT1
Abort, Retry, Ignore, Fail?
```

If this error message occurs, make sure that your printer is turned on and that the on-line light is lit.

Trap: *If your serial printer produces garbled text, your computer and printer are communicating at different speeds.*

Because your computer communicates with serial devices a bit at a time over a single wire, the two devices must agree on the format and speed that they will use to communicate. The speed that two devices use to communicate is called the *baud rate.* By default, DOS assigns a baud rate of 2400 bits per second to COM1. This approximates to 300 characters per second. However, your printer may be anticipating data at a faster or slower rate. Read the manual that accompanied your printer, or ask your computer retailer to determine the correct setting. Later in this chapter you will learn how to adjust the baud rate for your serial devices using the DOS MODE command.

In addition to the data transmission speed, a serial printer and computer must agree on several other data-communications parameters. Your printer manual will explain each of these parameters as well as what their settings should be. As you will find, the DOS MODE command allows you to assign to COM1 the values that your printer expects.

If you are using a parallel printer, you don't have to be concerned with these steps because your parallel printer should be immediately ready for use. If you have turned the printer on and cabled it correctly, it should print. If not, you may have a damaged cable or port.

In later chapters you will use the device names COM1 and LPT1 in conjunction with other DOS commands.

Configuring Your Devices

As just discussed, serial devices can require several preparatory steps before you can use them with DOS. This section will examine the DOS MODE command and the steps required to configure your serial port (COM1). Also, you will learn how MODE can enhance your screen and printer display.

With your Program Disk in drive A, issue the command

```
A> DIR  MODE.COM

Volume in drive A is MS330PP01
Directory of  A:\
```

```
MODE        COM        15440   7-24-87  12:00a
            1 File(s)     5120 bytes free
```

As you can see, MODE is an external DOS command. The next section examines how MODE can affect your screen display.

Controlling Your Screen Display

By default, your screen can display up to 25 lines of information of up to 80 characters in length (see Figure 5-10).

The MODE command allows you to change the display size of most monitors so that the monitor shows characters in enlarged print, with 25 lines each capable of containing 40 characters. To invoke this capability, enter the command

```
A> MODE 40
```

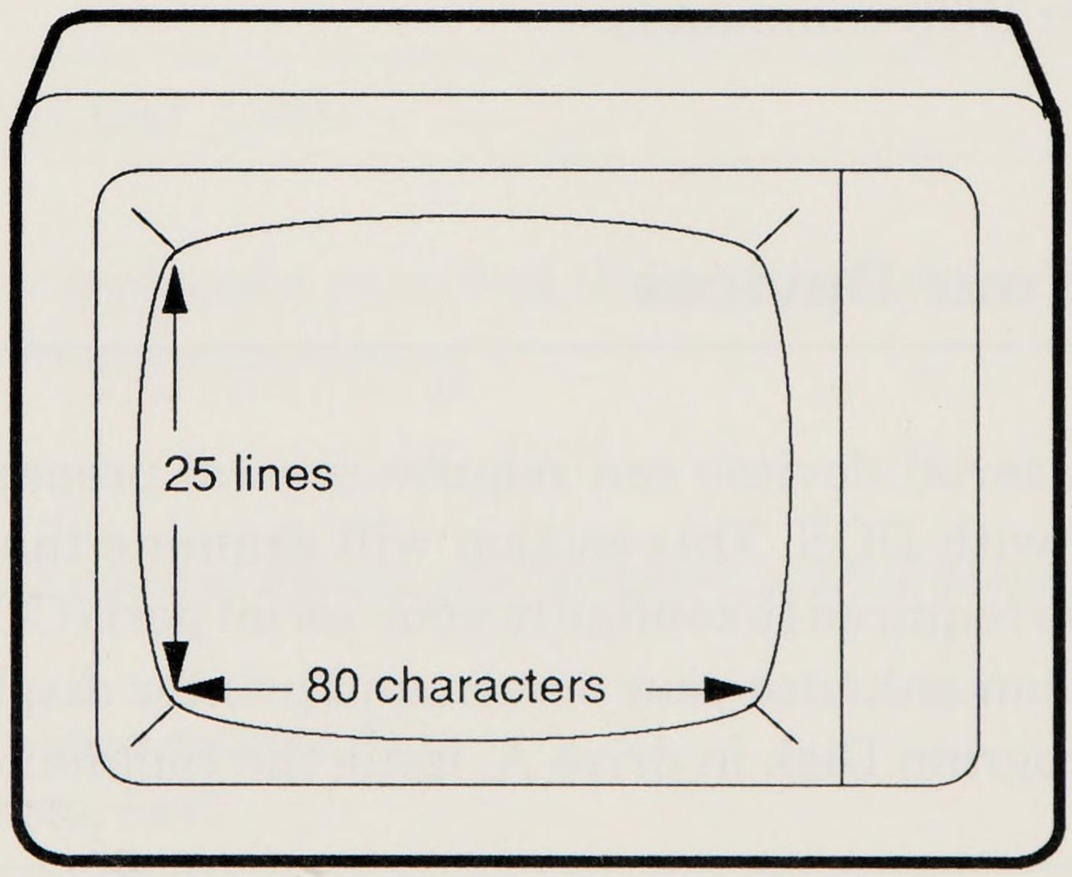


Figure 5-10. Default screen display

DOS will respond by clearing your screen display and placing an enlarged DOS prompt in the upper-left corner:

```
A>
```

If you issue a command such as

```
A> DIR
```

DOS will display your directory listing in this enlarged character size.

Trap: *Most software programs do not support a 40-column display.*

This enlarged character display may be easier to read than 80-column text, but most programs (such as your word processor) do not support this type of display. To restore your screen display to 80-column mode, enter the command

```
A> MODE 80
```

Tip: *You can use the DOS MODE command to connect two screens to your computer.*

Because some users have color screen displays while others have monochrome screens, programmers have to test their applications under each screen type in many cases to verify that the programs work correctly. Rather than having to purchase two computers, one with a monochrome screen and one with a color screen, DOS allows the programmers to connect both screens to their system, as shown in Figure 5-11.

When DOS boots, it will use the monochrome screen display. When the programmer needs to use the color screen display, the programmer simply issues the command

```
A> MODE CO80
```

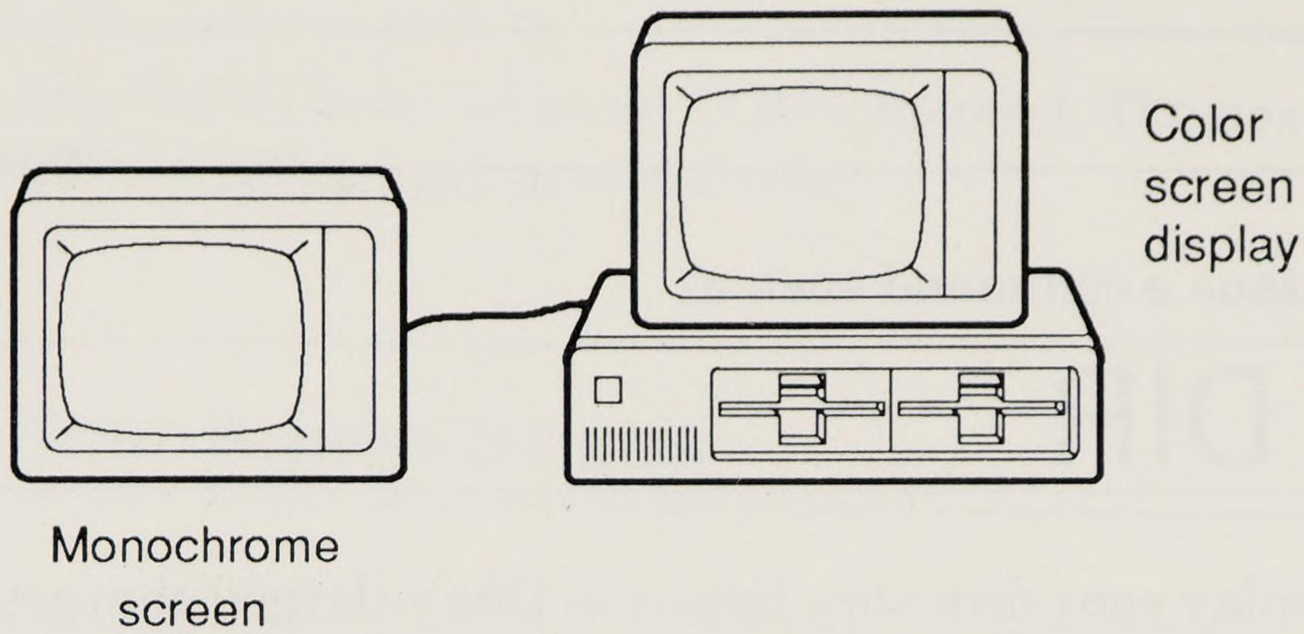


Figure 5-11. Monochrome and color monitors on same system

DOS will then display its output to the color screen display. To later select the monochrome system, the programmer simply enters the command

```
A> MODE MONO
```

The following table defines the values that you can issue in conjunction with the DOS mode command.

40	Specifies 40-column display
80	Specifies 80-column display
BW40	Specifies a 40-column black-and-white display
BW80	Specifies an 80-column black-and-white display
CO40	Specifies a 40-column color display
CO80	Specifies an 80-column color display
MONO	Selects a monochrome screen display

Trap: *Some screen displays do not correctly align text.*

Depending on the age and manufacturer of your screen, it may not align displayed characters correctly. The MODE command allows you to shift characters to the right or left as needed. To invoke this capability, use the MODE command

```
A> MODE 80,R,T
```

to move characters right one position, or

```
A> MODE 80,L,T
```

to move characters to the left. In both cases, DOS will display a test pattern on your screen and a prompt:

```
0123456789012345678901234567890123456789012345678901234567890123456789 0

Do you see the leftmost 0?
```

If you respond with **Y**, DOS will leave your screen unchanged. If you respond with **N**, DOS will shift your display one character position to the right. Once DOS makes this correction, all of your screen output will be correctly aligned. Most users will not require this form of the MODE command.

Trap: *Not all screen displays support the right and left character shift.*

When you issue the command

```
A> MODE 80,L,T
```

if MODE displays the message

```
Unable to shift screen right
```

your screen does not support character shifting. However, as stated above, most users will never require this capability.

Controlling Your Printer Output

By default, your parallel printer displays information with 80 characters per line and 6 vertical lines per inch. Using the STATES.DAT file,

```
A> COPY B:STATES.DAT
Arizona
Nevada
Washington
California
Texas
New York
^Z
        1 File(s) copied
```

the command

```
A> COPY B:STATES.DAT LPT1
```

results in the printout

Arizona
Nevada
Washington
California
Texas
New York

DOS also allows you to direct your printer to use 132 columns per line and 8 vertical lines per inch.

Tip: *The MODE command can change the output format of many printers allowing from 80 characters per line to 132 and from 6 lines per vertical inch to 8.*

To select 132 characters per line for your printed output, issue the command

```
A> MODE LPT1 132
```

DOS will respond with the message

```
LPT1: set for 132
```

To also select 8 lines per inch, use

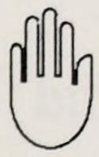
```
A> MODE LPT1 132,8
```

In this case, the command

```
A> COPY B:STATES.DAT LPT1
```

now results in a printout of the form

```
Arizona
Nevada
Washington
California
Texas
New York
```



Trap: *Not all printers support MODE.*

Although your computer may respond to the command

```
A> MODE LPT1 132,8
```

with the message

```
LPT1: set for 132
```

your printer may still print characters in 80-column mode. If this is the case, you may have to use an escape sequence (which will be discussed later) to change your printer setup. To restore your printer to normal type, simply issue a mode command that specifies 80 characters per line and 6 characters per inch.

Configuring Your Serial Device

As stated earlier in this chapter, your serial devices and your computer must agree on transmission speed and data-communication parameters before they can communicate. The DOS MODE command allows you to configure your computer's serial port to match the device specifications.

The format of the DOS MODE command for making necessary device changes is

```
MODE COM# baudrate[,parity[,databits[,stopbits[,p]]]]
```

where the following definitions apply.

- *COM#* specifies the serial port to configure (COM1, COM2, COM3, or COM4).
- *baudrate* specifies the transmission rate. Valid speeds are 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19200. Because each speed is unique, MODE requires you to enter only the first two digits of the baud rate. The default baud rate is 2400.
- *parity* specifies the device parity (the device manual will specify). Valid parity entries are "E" for even parity, "O" for odd parity, and "N" for no parity. The default value is "E".
- *databits* specifies the number of data bits in each transmission. Valid data-bit specifications are 7 and 8. The default value is 7.

- *stopbits* specifies the number of stop bits in each transmission. Valid stop-bit specifications are 1 and 2. The default for a 110-baud setting is 2. Otherwise, the default is 1.
- "p" specifies that DOS should attempt continuous transmission retries if the device times out (failing to acknowledge).

Don't panic! The format of the MODE command is actually quite straightforward. Items contained within brackets are optional. You can specify any of the following configurations:

```
A> MODE COM1 4800
A> MODE COM1 4800,O
A> MODE COM1 4800,E,8
A> MODE COM1 4800,E,8,2
A> MODE COM1 4800,E,8,2,P
```

Notice how the parameters correspond to the optional parameters of the MODE command just discussed.

Tip: *If you are connecting a device to your serial port, such as a laser printer, the documentation that accompanied your printer should specify the correct data-communication parameters.*

If you cannot locate the correct port configuration information in your device manual, ask your retailer for the correct settings.

Later chapters will examine other uses for the DOS MODE command.

Tips and Traps Summary

Creating Your Own Text Files

Tip: By using the device name CON as your source of information, you can create files with the COPY command.

Tip: The TYPE command displays the contents of a text file.

Trap: The DOS TYPE command cannot display all files. Files with the extension COM or EXE contain nonprintable characters that DOS cannot display on your screen.

Trap: If TYPE displays the error message

```
File not found
```

TYPE could not locate the file you specified for display.

Tip: In order to cancel a COPY CON FILENAME.EXT command, use the CTRL-C key combination.

Trap: COPY overwrites any existing file of the same name as your target file.

Tip: Use meaningful file names.

Using CON as an Output Device

Tip: For input operations, CON points to your keyboard. For output operations, CON points to your screen display.

Looking at DOS Device Names

Tip: Determine the type of printer that you are using, serial or parallel. The steps you must perform to use your computer are dependent upon the printer type.

Tips and Traps Summary (*continued*)

Tip: DOS names your parallel port LPT1 and your serial port COM1.

Tip: You can use the names LPT1 and PRN interchangeably to reference your computer's first parallel port.

Tip: You can use the names COM1 and AUX interchangeably to reference your computer's first serial port.

Using Your Printer

Trap: If DOS cannot access your printer, it will display the message

```
Not ready error writing device LPT1
Abort, Retry, Ignore, Fail?
```

Trap: If your serial printer produces garbled text, your computer and printer are communicating at different speeds.

Controlling Your Screen Display

Trap: Most software programs do not support a 40-column display.

Tip: You can use the DOS MODE command to connect two screens to your computer.

Trap: Some screen displays do not correctly align text.

Trap: Not all screen displays support the right and left character shift.

Tips and Traps Summary (*continued*)

Controlling Your Printer Output

Tip: The MODE command can change the output format of many printers allowing from 80 characters per line to 132 and from 6 lines per vertical inch to 8.

Trap: Not all printers support MODE.

Configuring Your Serial Device

Tip: If you are connecting a device to your serial port, such as a laser printer, the documentation that accompanied your printer should specify the correct data-communication parameters.

II

Putting DOS to Work For You

Chapters 1 through 5 laid the DOS foundation that new users will need to continue with the concepts presented in the second part of this book. Each of the remaining chapters examines a different DOS concept and presents tips and traps for users at all levels of DOS experience.

Keep in mind that, every DOS user has a unique level of understanding and that the tips and traps presented here attempt to cover the entire spectrum of DOS. Regardless of your current level of DOS experience, you should find new secrets and shortcuts in each chapter. Many of you will even find some new solutions that are quite fun.

6

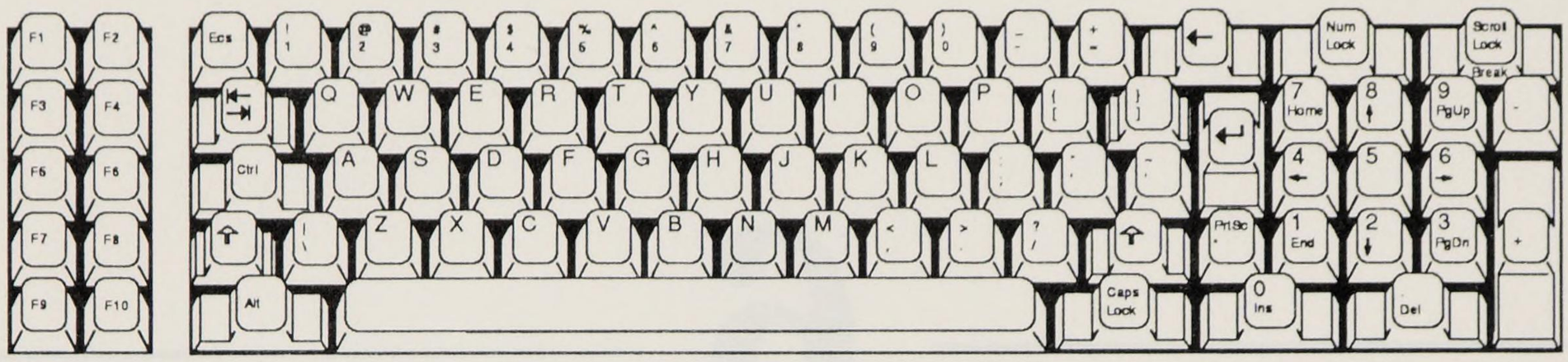
Getting the Most From Your Keyboard

Some people are always looking for ways to make their computers run faster so that they can be more productive during their computer sessions. One way to do this is to upgrade your system with an 80286 or 80386 processor so that you will get a faster response. Ironically, however, because of your computer's tremendous speed, most of you will have difficulty keeping your systems busy for only 25 percent of the time and your computer will sit idle most of the time, waiting for you to enter a response or command.

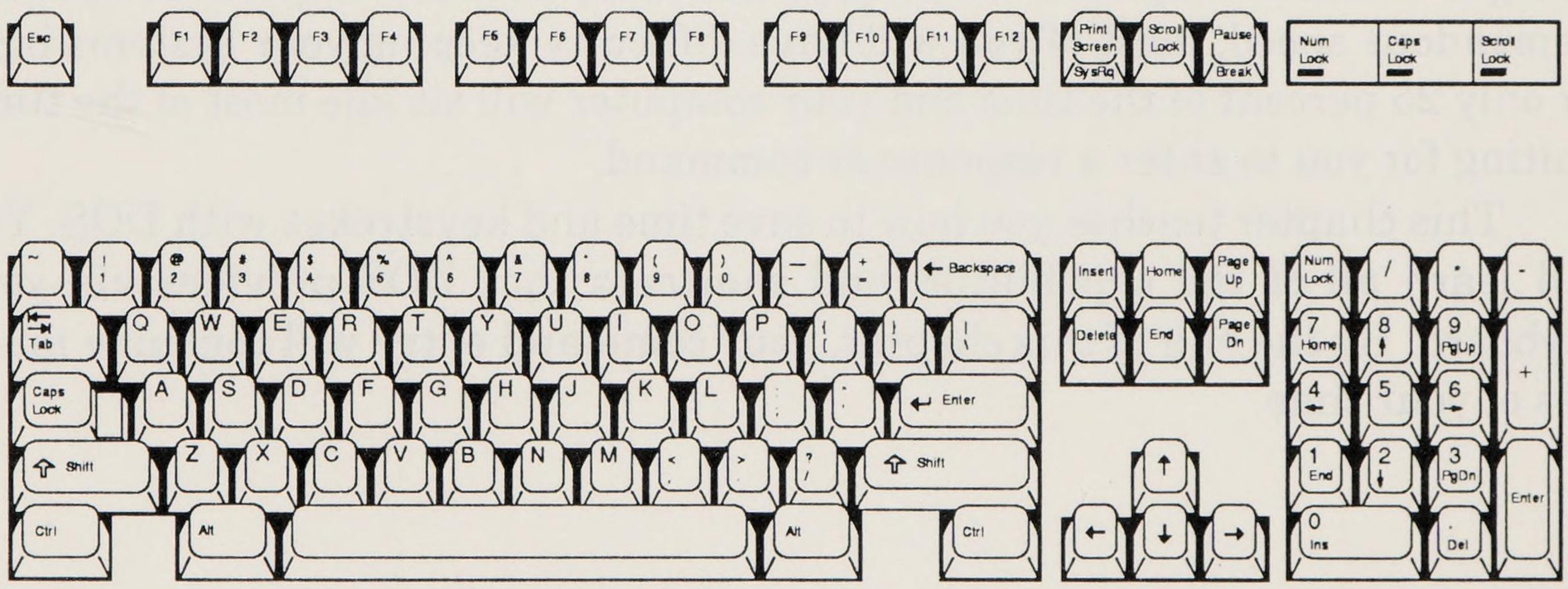
This chapter teaches you how to save time and keystrokes with DOS. You will learn all of the capabilities and shortcuts that DOS provides via your keyboard. By the end of this chapter, your command entry will consume much less of your time.

Common Keyboard Layouts

The following figures illustrate the keyboards commonly found on the IBM PC and PC-AT (see Figure 6-1). Although these keyboards differ slightly in appearance, their functionality for DOS users is identical.



IBM PC



IBM AT

Figure 6-1. Common keyboards for IBM PC and AT

This chapter will examine several DOS key combinations. You have already issued two of them in earlier chapters. In general, a *key combination* is a set of keys that you hold down at the same time. For example, the CTRL-C key combination directs you to hold down the key labeled CTRL and then press the key labeled C. Both keys are held down at the same time.

Restarting DOS with CTRL-ALT-DEL

In Chapter 1 you learned that you can restart DOS in one of two ways. First, you can simply turn off your computer's power, wait a few seconds, and turn the power back on. Second, you can press the CTRL-ALT-DEL key combination (see Figure 6-2).

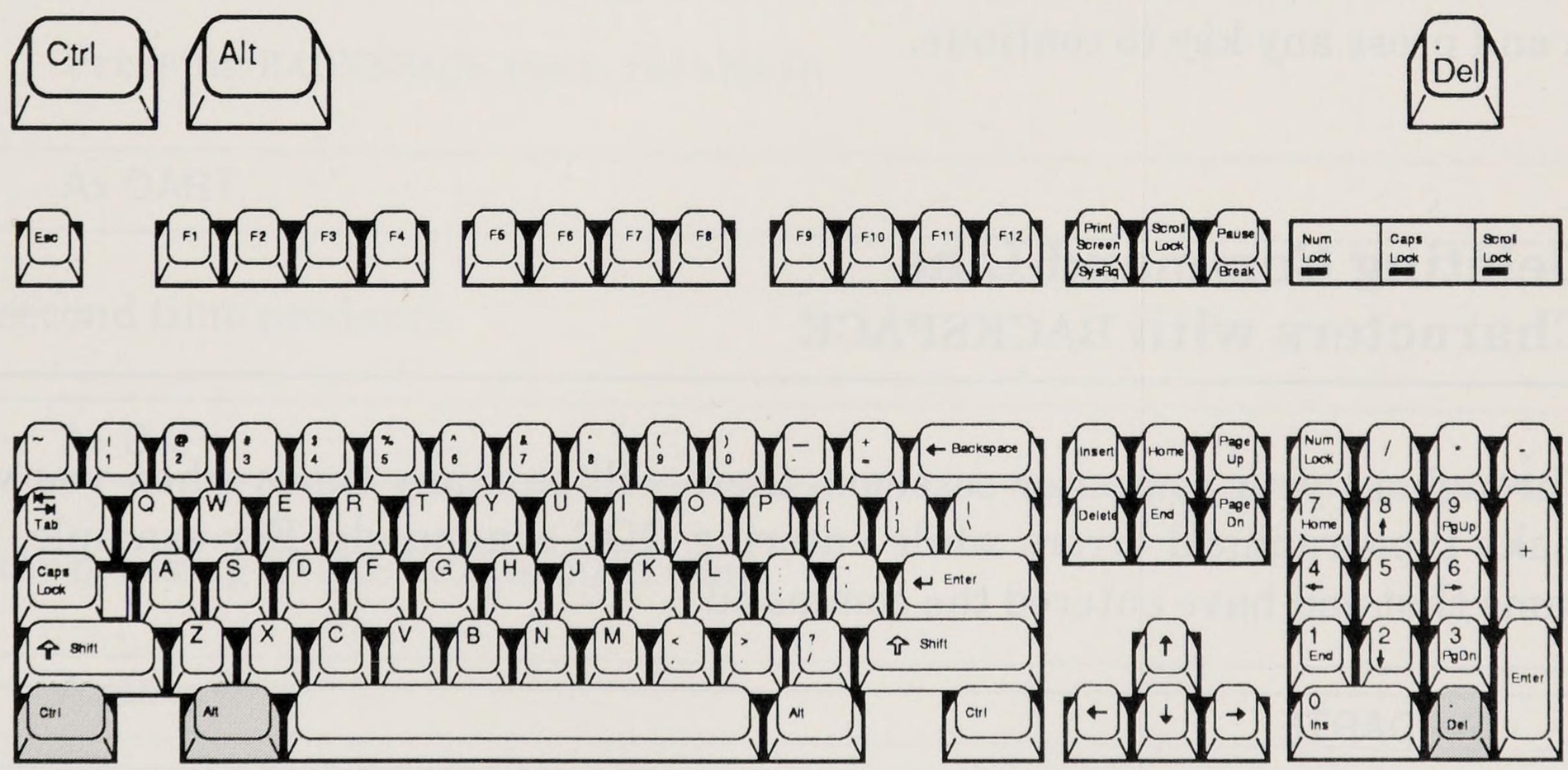


Figure 6-2. CTRL-ALT-DEL key combination

Trap: *If you restart DOS while a program is running, you risk losing all the files that the program was using.*

If possible, don't turn off your computer or restart DOS when you are not at the DOS prompt. Many programs open files to add or change information. If you restart DOS while these files are open, their contents will very likely be destroyed.

Trap: *DOS must be present in drive A, or drive A must be open (on fixed-disk systems) in order for DOS to restart.*

If your computer displays the message

```
Non-System disk or disk error
Replace and strike any key when ready
```

your DOS Program Disk is not in drive A. Insert your DOS Program Disk in drive A and press any key to continue. If you have DOS installed on a fixed disk (which will be discussed later in this book), open the disk-drive latch for drive A and press any key to continue.

Deleting Command-Line Characters with BACKSPACE

Throughout your computer sessions, there will be many times when you will make typographical errors while entering DOS commands. For example, assume that you have entered the command

```
A> DARTE
```

as opposed to the DOS DATE command. Assuming that you have not yet pressed the ENTER key, you can delete the letters in the command line by using the BACKSPACE key (as shown in Figure 6-3).

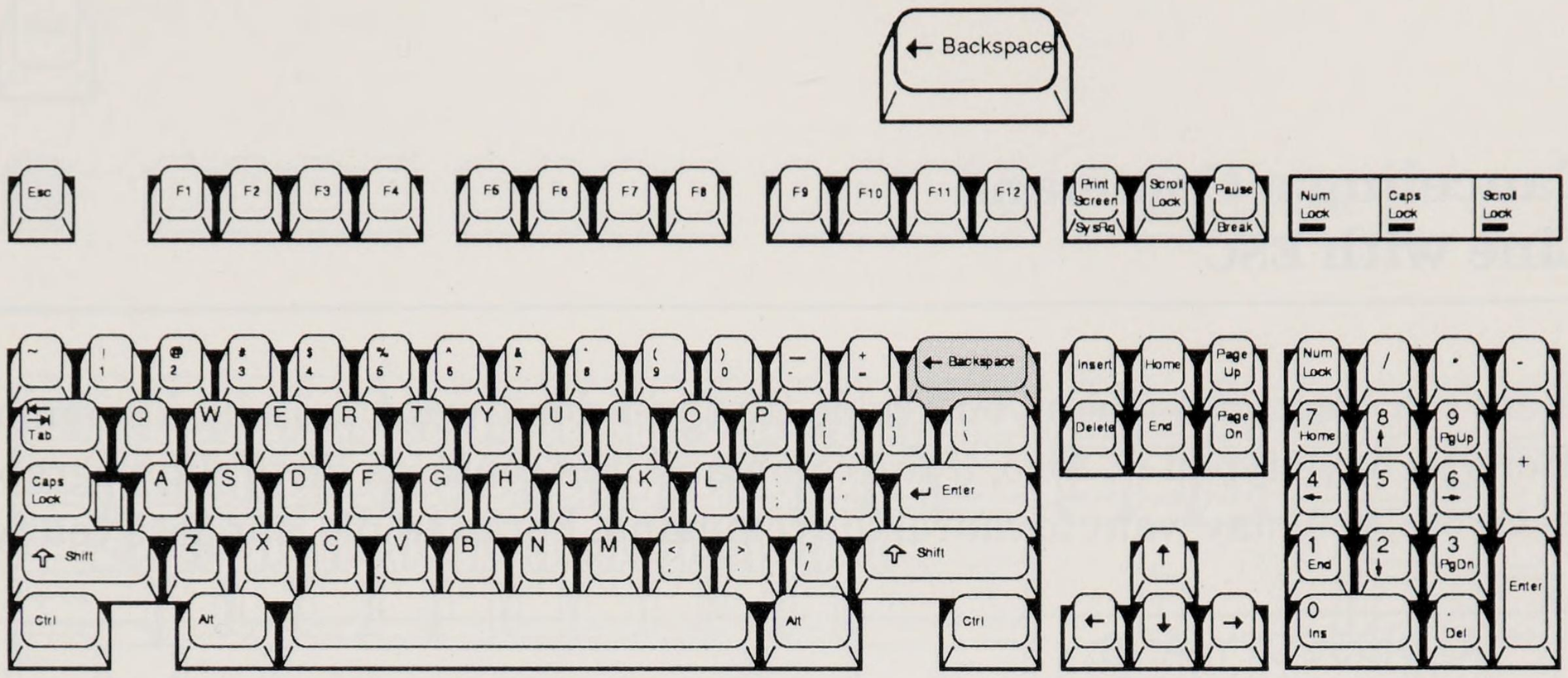


Figure 6-3. Location of BACKSPACE key

Pressing BACKSPACE once, results in

```
A> DART
```

A second time produces

```
A> DAR
```

After pressing BACKSPACE a third time

```
A> DA
```

you can type the correct command, as shown here:

```
A> DATE
```

Tip: *The BACKSPACE key directs DOS to delete the character that immediately precedes the cursor.*

Canceling a Command Line with ESC

There may be times when you decide that you don't want to issue a command after you have typed it. Also, if the command line contains several typographical errors, you may want to cancel the command. For example, suppose you type

```
A> DISSCOPY A: B:
```

As you know, the BACKSPACE key allows you to rub out or erase the characters that precede the cursor. However, if you have several characters to erase, or a command that you simply want to cancel, DOS provides a faster way. Locate the key labeled ESC on your keyboard (see Figure 6-4). If you press this key at the end of your command line, DOS will display a slash and move your cursor down one line, as shown here:

```
A> DISSCOPY A: B:\
_

A>
```

Next, press ENTER and DOS will ignore the command, redisplaying its prompt.

```
A> DISSCOPY A: B:\
_

A>
```

Tip: *The ESC key directs DOS to ignore the command line that precedes it.*

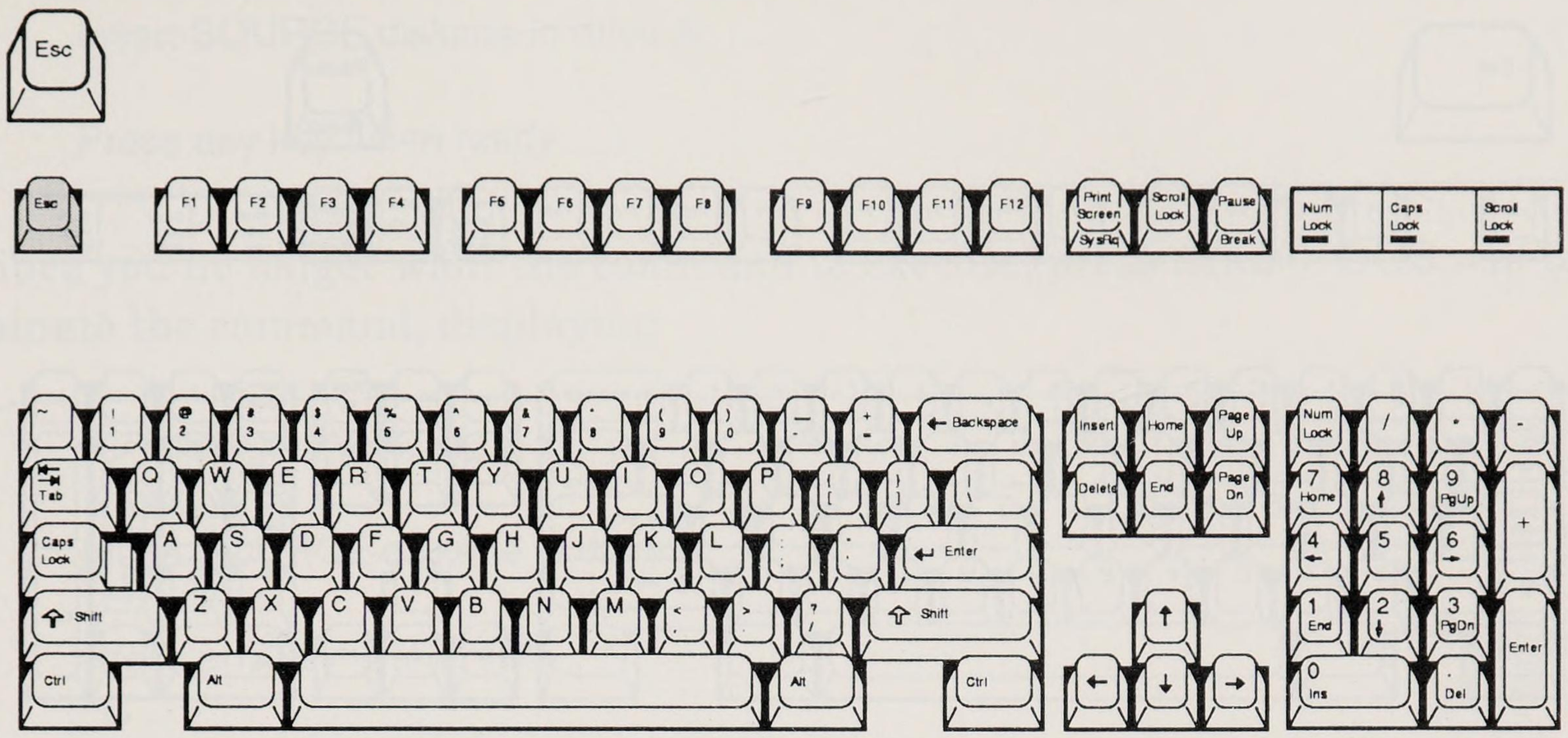


Figure 6-4. Location of the ESC key

If you don't press ENTER immediately following the ESC key, DOS allows you to retype in the command line.

```
A> DATE\
TIME
Current time is 19:12:09.93
Enter new time:
```

Terminating a Program with CTRL-BREAK

It is possible a program that you execute may contain an error (called a *bug*) that causes it to malfunction. In such cases, the program will no longer respond to you. Such programs are called *runaway programs*. Many times, the only way to terminate a runaway program is with the CTRL-BREAK key combination (see Figure 6-5).

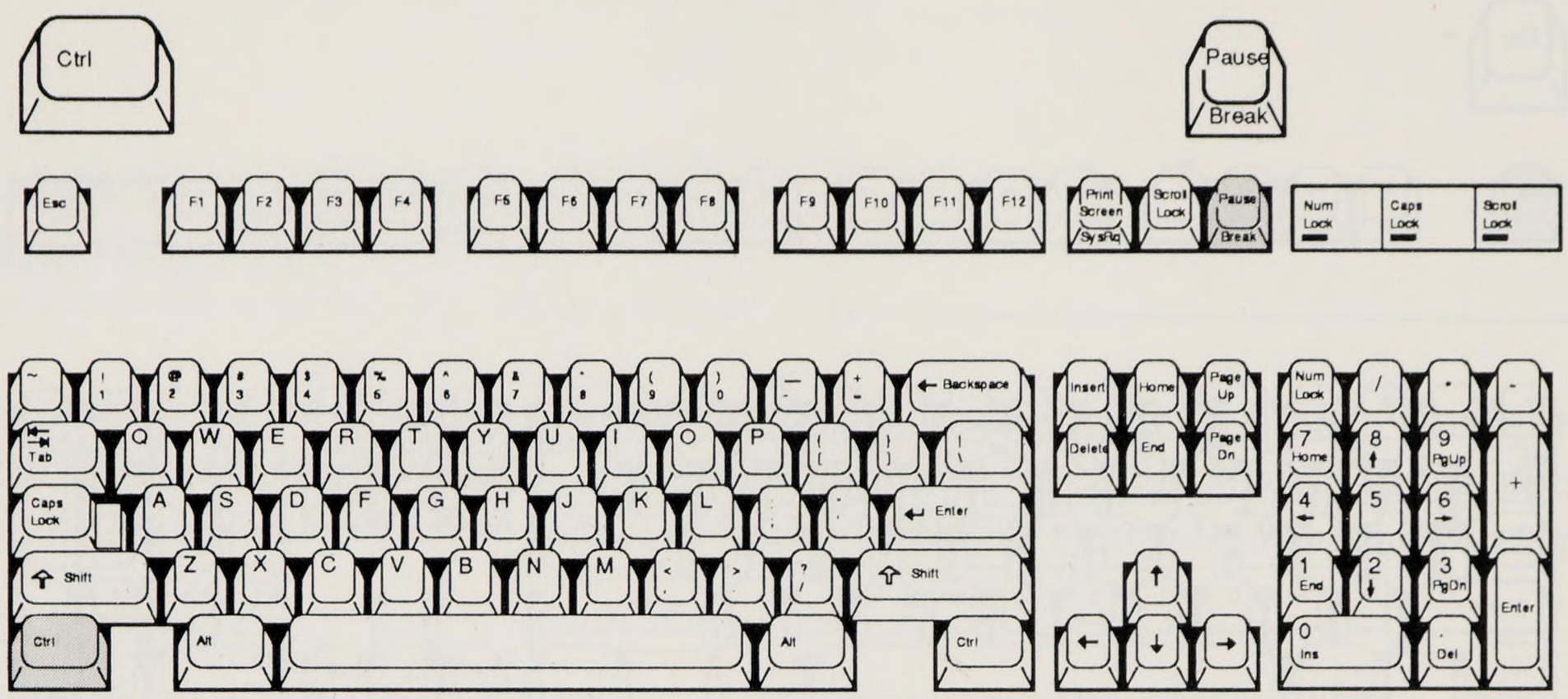


Figure 6-5. Location of the CTRL-BREAK key combination

If you want to cancel a DOS command, such as DISKCOPY, before it completes, the CTRL-BREAK key combination allows you to do that.

DOS allows you to use either CTRL-BREAK or CTRL-C to cancel a program. In Chapter 5 you used the CTRL-C key combination to terminate a file copy that you initiated from the CON device.

```
A> COPY CON NUMBERS.DAT
1
2
^C

A>
```

In a similar manner, you can use CTRL-C to terminate a program. Assume, for example, that you have invoked the DOS DISKCOPY command only to find that you don't have a blank target disk.

```
A> DISKCOPY  A:  B:

Insert SOURCE diskette in drive A:

Press any key when ready . . .
```

Since you no longer want the command to execute, press CTRL-C. DOS will terminate the command, displaying

```
A> DISKCOPY  A:  B:

Insert SOURCE diskette in drive A:

Press any key when ready . . .
^C

A>
```

Notice that DOS displays ^C on your screen display, indicating that it has terminated the command.

Tip: *In many cases, the CTRL-C key combination allows you to terminate the current command.*

This example uses CTRL-C to terminate the DOS LABEL command:

```
A> LABEL  B:

Volume in drive B is MISC

Volume label (11 characters, ENTER for none)? ^C

A>
```

Trap: *If you use CTRL-C to terminate a program that has open files, you may lose the contents of those files.*

As stated, many programs open files to add or change information. If you terminate the program while these files are open, their contents will very likely be destroyed. Therefore, use CTRL-C only as a last resort to terminate a command.

Tip: *You can use the key combinations CTRL-C and CTRL-BREAK interchangeably to terminate a program.*

Remember that CTRL-C and CTRL-BREAK function identically.

Trap: *CTRL-C cannot terminate all programs.*

If you are unable to terminate a runaway program with CTRL-C, restarting DOS is your only alternative.

Suspending Output with CTRL-S

When you issue the DOS directory command, it is possible for files names to scroll past you on the screen so fast that you miss the desired file. In Chapter 3 you learned that DIR provides the /P qualifier to suspend the output of your directory listing with each screenful of information displayed.

```
A> DIR /P

Volume in drive A is MS330PP01
Directory of A:\

4201        CPI   17089   7-24-87  12:00a
5202        CPI   459     7-24-87  12:00a
ANSI        SYS   1647    7-24-87  12:00a
APPEND      EXE   5794    7-24-87  12:00a
ASSIGN      COM   1530    7-24-87  12:00a
ATTRIB      EXE   10656   7-24-87  12:00a
CHKDSK      COM   9819    7-24-87  12:00a
COMMAND     COM   25276   7-24-87  12:00a
```

```
COMP        COM   4183   7-24-87  12:00a
COUNTRY     SYS   11254  7-24-87  12:00a
DISKCOMP    COM   5848   7-24-87  12:00a
DISKCOPY    COM   6264   7-24-87  12:00a
DISPLAY     SYS   11259  7-24-87  12:00a
DRIVER      SYS   1165   7-24-87  12:00a
EDLIN       COM   7495   7-24-87  12:00a
EXE2BIN     EXE   3050   7-24-87  12:00a
FASTOPEN    EXE   3888   7-24-87  12:00a
FDISK       COM   48919  7-24-87  12:00a
FIND        EXE   6403   7-24-87  12:00a
FORMAT      COM   11671  7-24-87  12:00a
GRAFTABL    COM   6136   7-24-87  12:00a
GRAPHICS    COM   13943  7-24-87  12:00a
JOIN        EXE   9612   7-24-87  12:00a
Strike a key when ready . . .
```

However, most DOS commands do not provide a qualifier such as /P to suspend output. Instead, you can temporarily suspend the output of a command by pressing the CTRL-S key combination (see Figure 6-6).

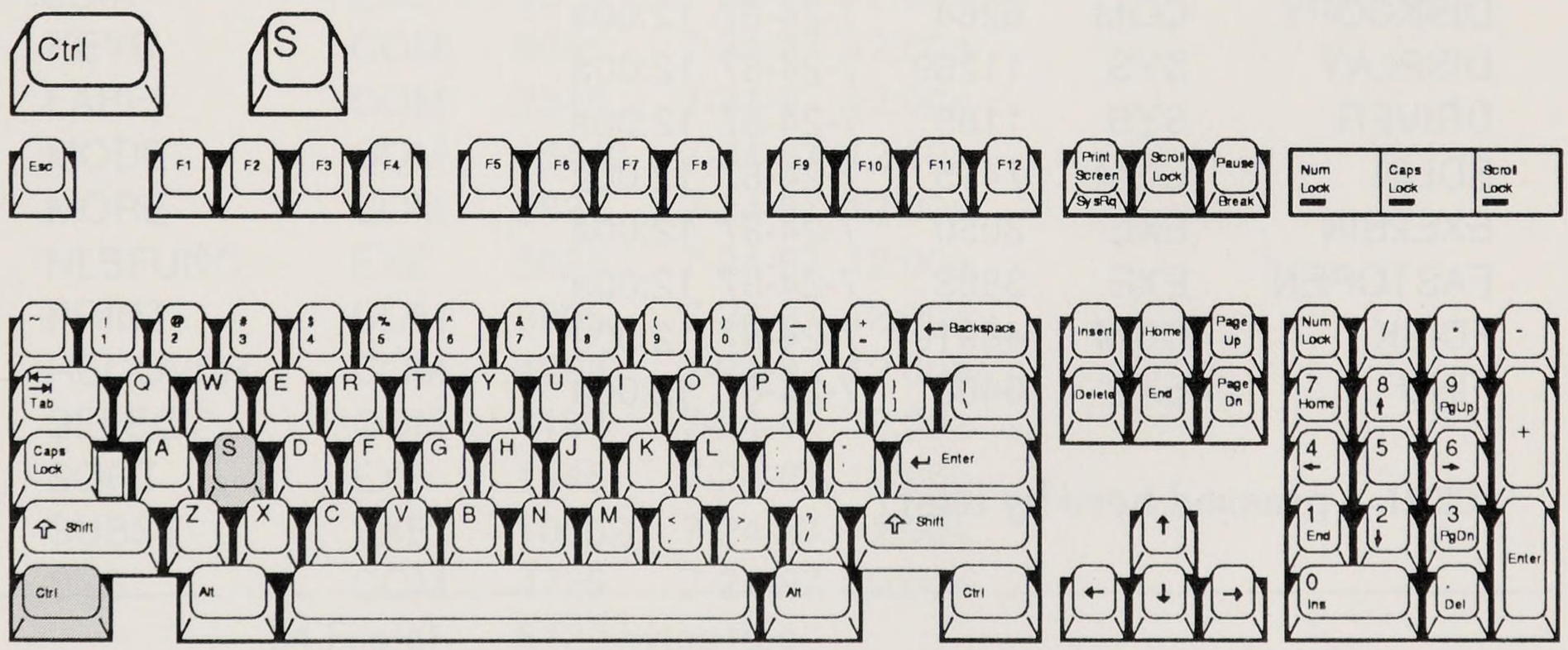


Figure 6-6. Location of the CTRL-S key combination

Tip: *You can suspend the output of a command temporarily by pressing the CTRL-S key combination. To resume output, press any key.*

When you issue the directory command

```
A> DIR
```

the file names begin to scroll past you on the screen. You can suspend output by pressing CTRL-S.

```
Volume in drive A is MS330PP01
Directory of  A:\

4201        CPI     17089   7-24-87  12:00a
5202        CPI     459     7-24-87  12:00a
ANSI        SYS     1647    7-24-87  12:00a
APPEND      EXE     5794    7-24-87  12:00a
ASSIGN      COM     1530    7-24-87  12:00a
ATTRIB      EXE     10656   7-24-87  12:00a
CHKDSK      COM     9819    7-24-87  12:00a
COMMAND     COM     25276   7-24-87  12:00a
COMP        COM     4183    7-24-87  12:00a
COUNTRY     SYS     11254   7-24-87  12:00a
DISKCOMP    COM     5848    7-24-87  12:00a
DISKCOPY    COM     6264    7-24-87  12:00a
DISPLAY     SYS     11259   7-24-87  12:00a
DRIVER      SYS     1165    7-24-87  12:00a
EDLIN       COM     7495    7-24-87  12:00a
EXE2BIN     EXE     3050    7-24-87  12:00a
FASTOPEN    EXE     3888    7-24-87  12:00a
FDISK       COM     48919   7-24-87  12:00a
FIND        EXE     6403    7-24-87  12:00a

(CTRL-S pressed here by user)
```

Once you have examined the files displayed, press any key to resume the output of the directory listing (see Figure 6-7).

```
Volume in drive A is MS330PP01
Directory of  A:\

4201        CPI     17089   7-24-87  12:00a
5202        CPI     459     7-24-87  12:00a
ANSI        SYS     1647    7-24-87  12:00a
APPEND      EXE     5794    7-24-87  12:00a
ASSIGN      COM     1530    7-24-87  12:00a
ATTRIB      EXE     10656   7-24-87  12:00a
CHKDSK      COM     9819    7-24-87  12:00a
COMMAND     COM     25276   7-24-87  12:00a
COMP        COM     4183    7-24-87  12:00a
COUNTRY     SYS     11254   7-24-87  12:00a
DISKCOMP    COM     5848    7-24-87  12:00a
DISKCOPY    COM     6264    7-24-87  12:00a
DISPLAY     SYS     11259   7-24-87  12:00a
DRIVER      SYS     1165    7-24-87  12:00a
EDLIN       COM     7495    7-24-87  12:00a
EXE2BIN     EXE     3050    7-24-87  12:00a
FASTOPEN    EXE     3888    7-24-87  12:00a
FDISK       COM     48919   7-24-87  12:00a
FIND        EXE     6403    7-24-87  12:00a
FORMAT      COM     11671   7-24-87  12:00a
GRAFTABL    COM     6136    7-24-87  12:00a
GRAPHICS    COM     13943   7-24-87  12:00a
JOIN        EXE     9612    7-24-87  12:00a
KEYB        COM     9041    7-24-87  12:00a
LABEL       COM     2346    7-24-87  12:00a
MODE        COM     15440   7-24-87  12:00a
MORE        COM     282     7-24-87  12:00a
NLSFUNC     EXE     3029    7-24-87  12:00a
PRINT       COM     8995    7-24-87  12:00a
RECOVER     COM     4268    7-24-87  12:00a
SELECT      COM     4132    7-24-87  12:00a
SORT        EXE     1946    7-24-87  12:00a
SUBST       EXE     10552   7-24-87  12:00a
SYS         COM     4725    7-24-87  12:00a
          34 File(s)     5120 bytes free
```

Figure 6-7. Resuming output by pressing any key

This same technique works when you use the DOS TYPE command to display the contents of a large file.

```
A> TYPE CHAPTER5.TXT
```

Printing the Contents of Your Current Screen with SHIFT-PRTSC

In Chapter 5 you learned how to print files by using the command

```
A> COPY FILENAME.EXT LPT1
```

At that time, serial printer users found that they had to issue the command

```
A> COPY FILENAME.EXT COM1
```

to print their files. When you want DOS to print the current contents of your screen display, do so using the SHIFT-PRTSC key combination.

Tip: *The SHIFT-PRTSC key combination directs DOS to print the contents of the current screen.*

With your Program Disk in drive A, issue the command

```
A> DIR *.SYS

Volume in drive A is MS330PP01
Directory of A:\

ANSI        SYS    1647    7-24-87  12:00a
COUNTRY     SYS    11254   7-24-87  12:00a
DISPLAY     SYS    11259   7-24-87  12:00a
DRIVER      SYS    1165    7-24-87  12:00a
          4 File(s)    5120 bytes free
```

Next, press the SHIFT-PRTSC key combination (see Figure 6-8). If your printer is on-line, DOS will print the current screen contents.

Trap: *By default, DOS uses LPT1 as its printer port. Serial printer users must direct DOS to use COM1.*

If you have a serial printer, the SHIFT-PRTSC operation just discussed will have no effect because DOS assumes that your printer is connected to LPT1. Unless you tell DOS otherwise, it has no way of knowing about your serial printer. You need to use the DOS MODE command to direct DOS to use your serial printer.

Tip: *Since, by default, DOS uses the LPT1 port for your printer output, the DOS MODE command allows you to direct DOS to use COM1.*

If you have a serial printer, place your Program Disk in drive A and issue the command

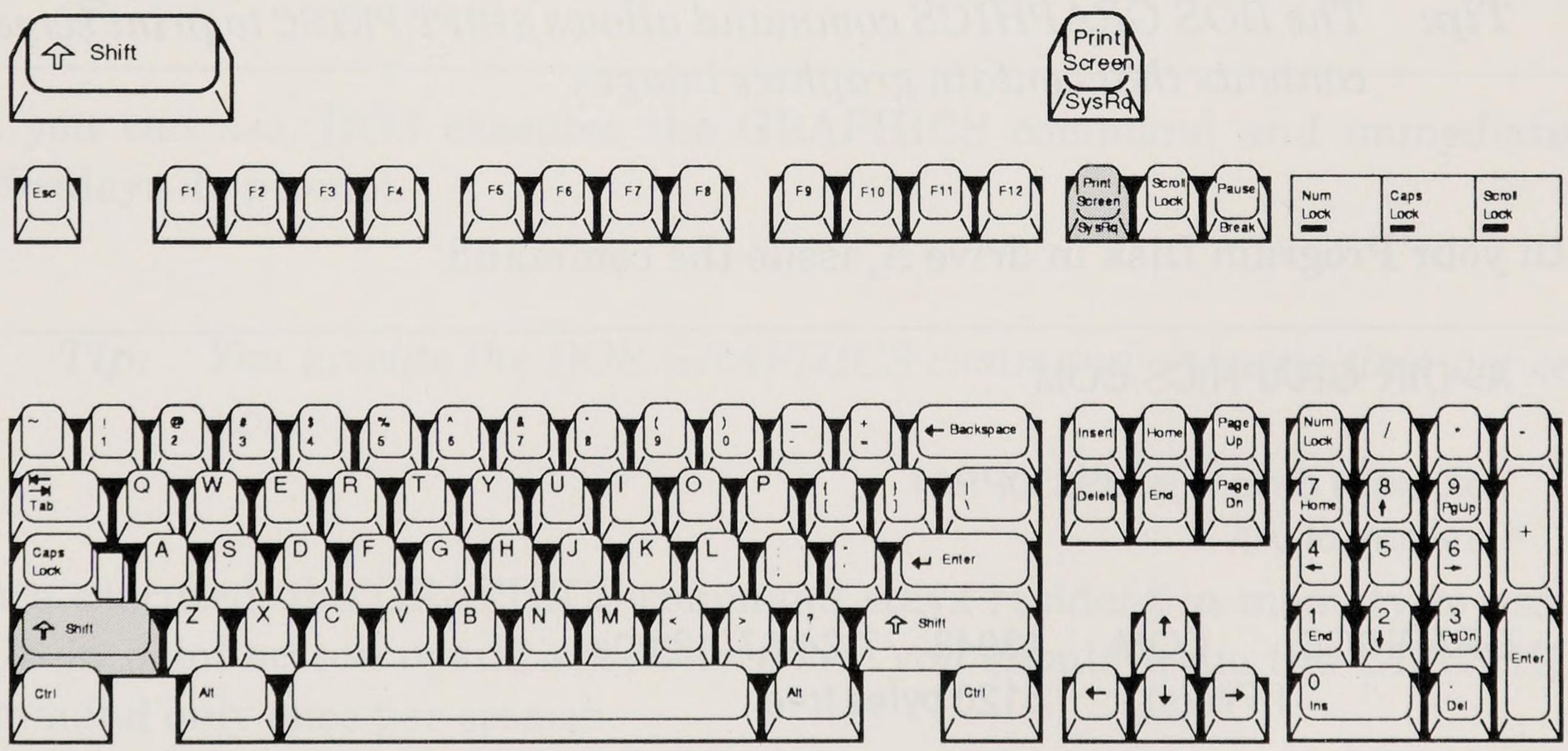


Figure 6-8. Location of SHIFT-PRTSC key combination

```
A> MODE  LPT1:=COM1
```

MODE will respond with

```
Resident portion of MODE loaded

LPT1: rerouted to COM1:
```

Again, try the SHIFT-PRTSC key combination to print your current screen contents. In this case, assuming your printer is on-line, the operation should be successful. If you are using a parallel printer, you can ignore this feature of the MODE command.

Trap: *By default, SHIFT-PRTSC prints only text, not graphics.*

All of the examples examined thus far have dealt strictly with screens containing text as opposed to graphics. If you want to print the contents of your screen when it contains graphics—such as a spreadsheet plot, seen in Figure 6-9—you must first issue the DOS GRAPHICS command.

Tip: *The DOS GRAPHICS command allows SHIFT-PRTSC to print screen contents that contain graphics images.*

With your Program Disk in drive A, issue the command

```
A> DIR  GRAPHICS.COM

Volume in drive A is MS330PP01
Directory of  A:\

GRAPHICS      COM      13943   7-24-87  12:00a
            1 File(s)      5120 bytes free
```

GRAPHICS is an external DOS command. The only time you need to invoke this command is when you want to print graphics screens with SHIFT-PRTSC.

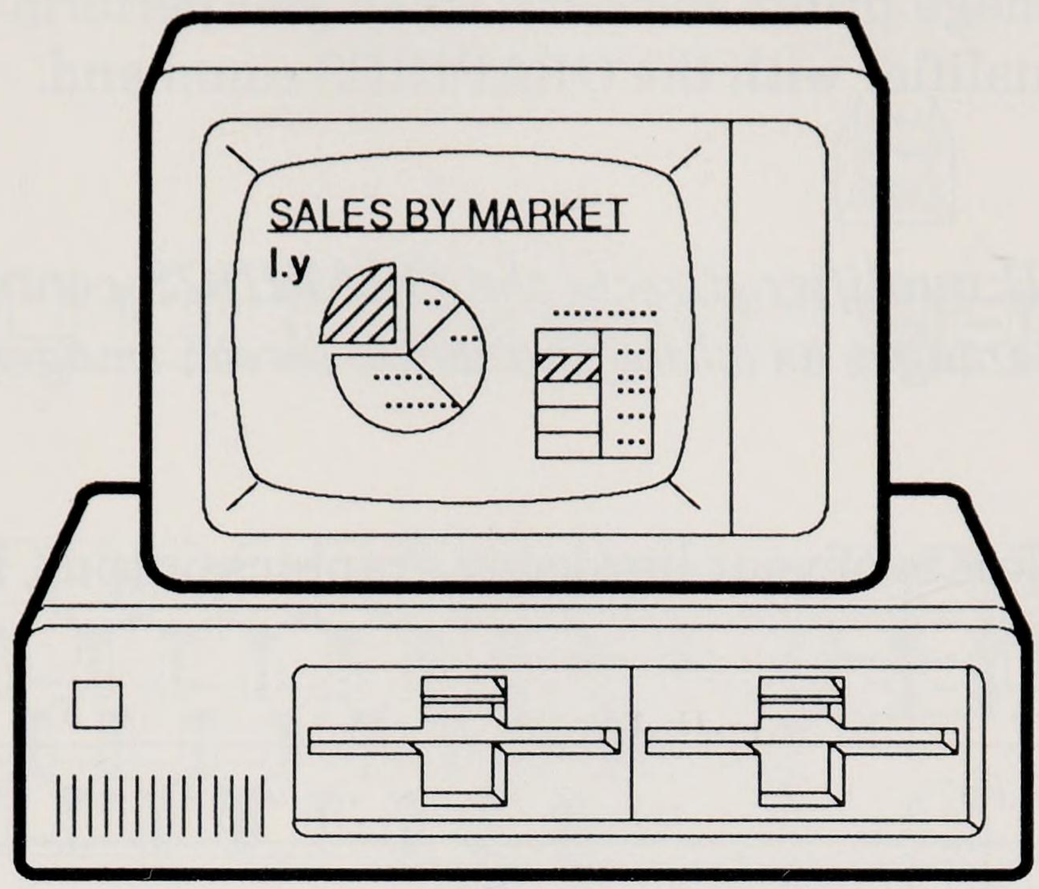


Figure 6-9. Spreadsheet graphics

To invoke GRAPHICS, type

```
A> GRAPHICS
A>
```

As you can see, DOS executes the GRAPHICS command and immediately redisplays its prompt.

Tip: *You execute the DOS GRAPHICS command only one time per session.*

Once executed, the GRAPHICS command stays resident in memory to assist DOS in print-screen operations. Therefore, you should issue the GRAPHICS command only once per session.

Trap: *The printout of your graphics image may be too dark.*

If your graphics image prints too dark when you perform SHIFT-PRTSC operations, use the /R qualifier with the GRAPHICS command.

Tip: *The /R qualifier directs the GRAPHICS command to print dark screen images as white and white screen images as dark.*

To reverse the darkness of your hardcopy graphics output, invoke the GRAPHICS command as

```
A> GRAPHICS /R
```

Remember that the only time you need to invoke GRAPHICS is when you are using SHIFT-PRTSC to print graphics screen images.

Echoing Text to Your Printer With CTRL-PRTSC

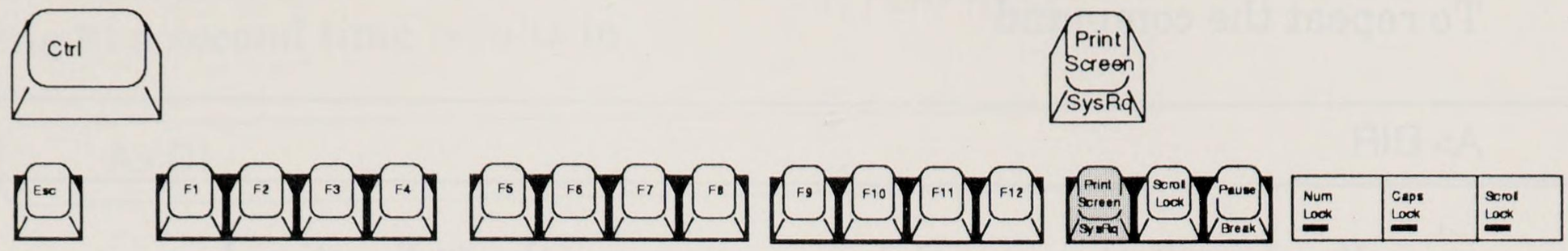
Just as the SHIFT-PRTSC key combination directs DOS to print the current contents of your screen display, CTRL-PRTSC directs DOS to send each character that it writes to your screen to your printer as well (see Figure 6-10).

Tip: *The CTRL-PRTSC key combination directs DOS to send the characters that it writes to your screen display to your printer as well.*

With your printer on-line, press CTRL-PRTSC. Nothing should happen. Next, issue the DOS DIR command.

```
A> DIR
```

As DOS writes your directory listing to the screen display, it also sends it to your printer. To turn this *character echoing* off, press CTRL-PRTSC again.



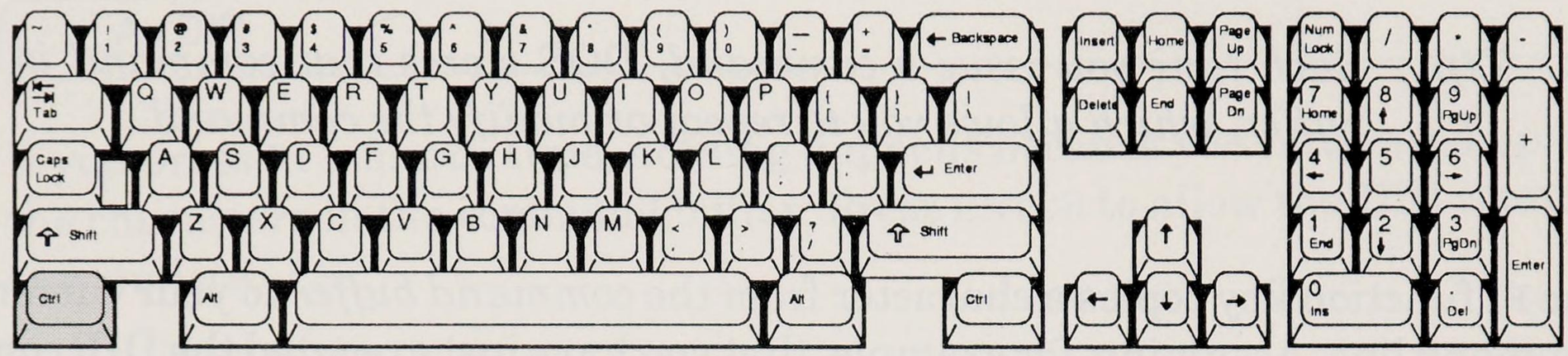


Figure 6-10. Location of CTRL-PRTSC key combination

Tip: *The CTRL-PRTSC key combination works as a toggle. The first time you press it, DOS enables character echo to your printer. The second time you press it, DOS disables character echo.*

Using the DOS Function Keys

Locate the DOS function keys on your keyboard (see Figure 6-11). These keys are so named because your software programs often assign specific functions to them. As you will see, DOS is no exception. You have already used the F6 function key to place an end-of-file marker in the files that you copy from your keyboard, as shown here:

```
A> COPY CON FILENAME.EXT
This is line one.
This is line two.
^Z
                1 File(s) copied
```

This section examines the rest of the DOS function keys.
To repeat the command

```
A> DIR
```

once your directory listing completes, you can use the F1 function key.

Tip: *Each time you issue a command, DOS stores that command in a buffer, which allows you to repeat or modify the command.*

The F1 function key copies a character from the *command buffer* to your current command line. Assuming, for example, that you have just executed the DIR command, pressing F1 once results in

```
A> D
```

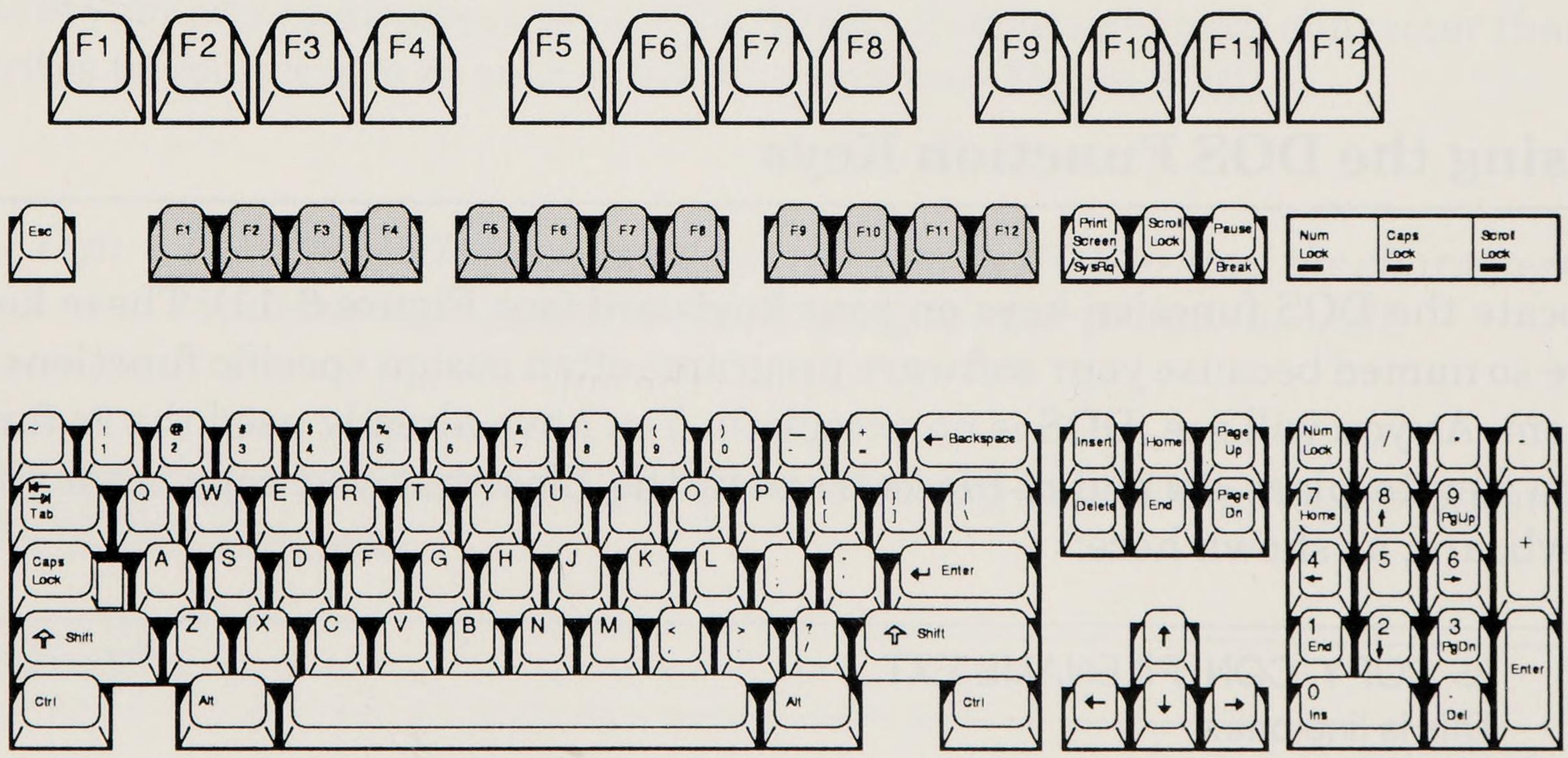


Figure 6-11. Location of the function keys

This makes sense because "D" is the first letter in the command buffer. Pressing F1 a second time results in

```
A> DI
```

Likewise, pressing F1 again yields

```
A> DIR
```

If you press F1 a fourth time, nothing happens because you have used up all of the characters in the command buffer. Press ENTER to allow the DIR command to complete.

Tip: *The F1 function key copies a single character from the command buffer for use in your current command line.*

Once DIR completes, the third and easiest way to repeat the command is to use the F3 function key. Pressing F3 from the DOS prompt results in

```
A> DIR
```

Tip: *The F3 function key copies the entire command buffer to your command line.*

The F3 function key directs DOS to copy the entire command buffer to your current command line. You are then free to execute the command or to type additional characters on your command line, as shown here:

```
A> DIR COMMAND.COM
```

Trap: *DOS keeps track of the number of characters in your current command line and copies only the characters past that point from the previous command buffer.*

Assume that your previous command is

```
A> DIR FILENAME.EXT
```

If you type

```
A> DEL
```

and then press F3, DOS will ignore the first three characters in the command buffer, resulting in

```
A> DEL FILENAME.EXT
```

Editing Text in the Command Buffer

As you just saw, the DOS function keys allow you to quickly repeat DOS commands. In a similar manner, you may want to slightly modify (edit) the previous command before you issue it. As you will see, the DOS function keys are very helpful in this process.

Assume, for example, that you have misspelled DISKCOPY.

```
A> DISSCOPY A: B:
Bad command or file name

A>
```

If you press F1 three times, DOS will display

```
A> DIS
```

You can now type **K**.

```
A> DISK
```

Next, press F3 and DOS will display

```
A> DISKCOPY A: B:
```

Your command is now ready to execute.

As another example, if you have entered the command as

```
A> DISCOPY A: B:
```

forgetting the "K", you can perform a similar operation. As before, press F1 three times.

```
A> DIS
```

Next, locate the INS key on your keyboard (see Figure 6-12) and press it. Nothing will appear to change on your screen. Type **K**.

```
A> DISK
```

Now press F3.

```
A> DISKCOPY A: B:
```

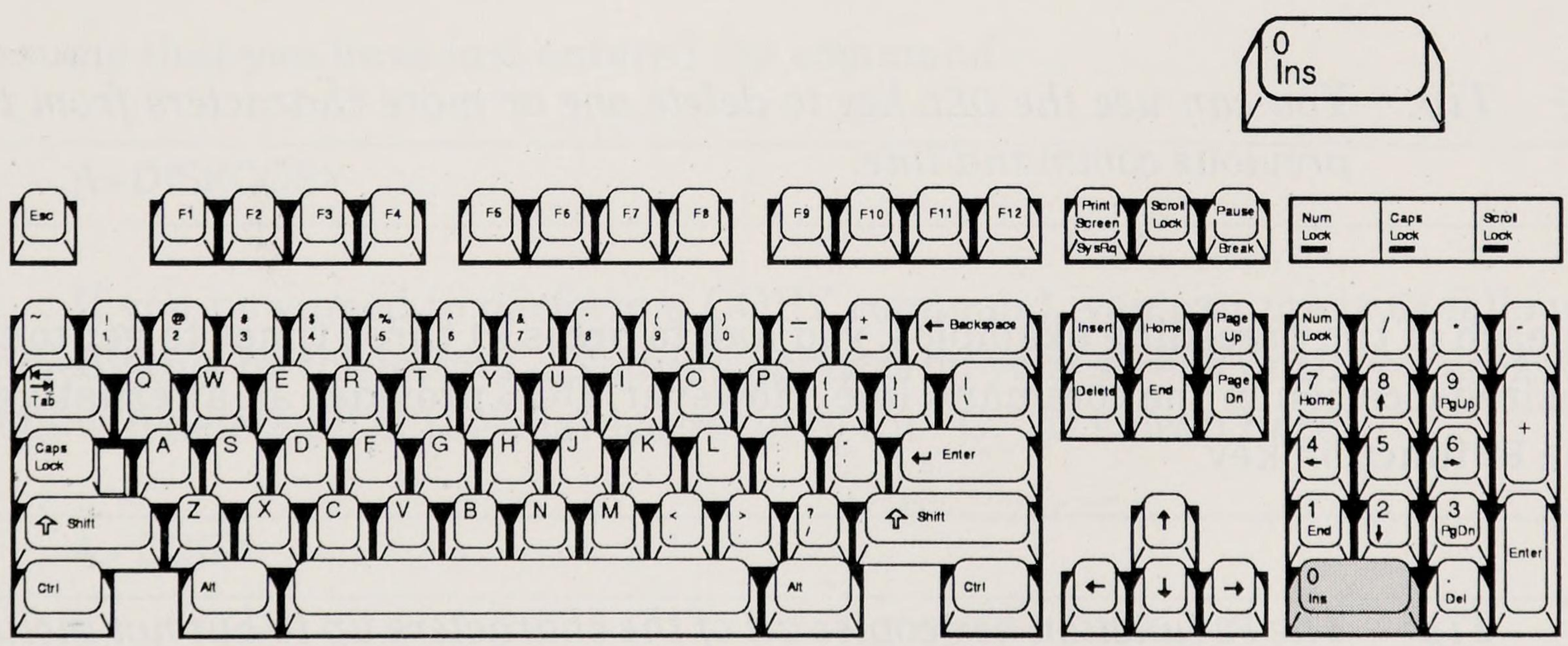


Figure 6-12. Location of INS *key*

As you can see, DOS has inserted the letter "K" as desired.

Tip: *The INS key allows you to insert one or more characters into the previous command line. INS works as a toggle: the first press enables insert mode; the second press enables overstrike mode.*

If you need to delete a character from your previous command line,

```
A> DISKKCOPY A: B:
Bad command or file name

A>
```

you can use the DEL key. In this case, to delete the first "K," press F1 three times.

```
A> DIS
```

Next, press DEL key and then F3.

```
A> DISKCOPY A: B:
```

As you can see, DOS has deleted the extra "K" as desired.

Tip: *You can use the DEL key to delete one or more characters from the previous command line.*

In each of the preceding examples, you had to press F1 three times to get to the desired location in the command line. However, DOS provides an alternative—the F2 function key.

Tip: *The F2 function key copies all of the characters up to but not including the character that you type immediately after pressing F2.*

Assume that you have just issued the command

```
A> DISKCCOPY  A:  B:
Bad command or file name

A>
```

Rather than pressing F1 several times, press F2 and immediately type **C**. DOS will display

```
A> DISK
```

Next, press DEL as just discussed and then F3. DOS will display the corrected command, as shown here:

```
A> DISKCOPY  A:  B:
```

The F4 function key works in just the opposite manner.

Tip: *The F4 function key directs DOS to delete all of the characters in the previous command line preceding the character that you type immediately after pressing F4.*

Assume that you have just entered the command

```
A> DISKCOPY
```

If you now need to perform a COPY command, you can press F4 followed immediately by the letter **C**. Nothing appears on your screen until you press F3. DOS will delete all of the characters up to the "C," as shown here:

```
A> COPY
```

If you want to edit your current command line, use the F5 key.

Tip: *The F5 function key allows you to edit your current command line.*

Suppose, for example, that before pressing ENTER to invoke a command, you realize that your command line contains an error.

```
A> DISSCOPY A: B:
```

Press F5. DOS will place the cursor immediately below the first letter in your command line, as shown here:

```
A> DISSCOPY A: B:@
_
```

Use the keys discussed throughout this section to edit your command line. For example, pressing F1 three times yields

```
A> DISSCOPY A: B:@
DIS
```

Next, type **K** and press F3.

```
A> DISSCOPY A: B:@
DISKCOPY A: B:
```

Your command is now ready to execute. Simply press ENTER.

Enhancing Your Files with ALT

Many users are familiar with use of the ALT key in the CTRL-ALT-DEL key combination that restarts DOS. However, most users don't know that you can use ALT to enhance your text files.

Assume, for example, that you have been keeping DOS notes about each command throughout this book:

DISKCOPY.NTS

FORMAT.NTS

CLS.NTS

LABEL.NTS

and so on. In your FORMAT.NTS file, you want to add the following message:

```
WARNING!

FORMAT destroys the contents of your target disk!
```

By using the ALT key, you can enhance your note so that it reads

```
WARNING!
```

To do so, you must use unique character values called *extended character*. These characters, along with the standard set of characters your computer uses, are shown in Table 6-1.

To enter the character

```
┌
```

hold down the ALT key and use the numeric keypad to the right of your keyboard (not the upper row of number keys). First hold down the ALT key and type **218**. When you release the ALT key, DOS will display

```
A> COPY CON FORMAT.NTS
┌
```

Next, type **196** nine times.

```
A> COPY CON FORMAT.NTS
┌──────
```

Then type **191** to create the upper-right corner.

Table 6-1. Standard Set of Characters

Number	Character	Number	Character	Number	Character
0	Null	30	RS	60	<
1	☺	31	US	61	=
2	☻	32	Space	62	>
3	♥	33	!	63	?
4	♦	34	"	64	@
5	♣	35	#	65	A
6	♠	36	$	66	B
7	Bell	37	%	67	C
8	BS	38	&	68	D
9	HT	39	'	69	E
10	LF	40	(	70	F
11	VT	41	)	71	G
12	FF	42	*	72	H
13	CR	43	+	73	I
14	SO	44	,	74	J
15	SI	45	-	75	K
16	DLE	46	.	76	L
17	DC1	47	/	77	M
18	DC2	48	0	78	N
19	DC3	49	1	79	O
20	DC4	50	2	80	P
21	NAK	51	3	81	Q
22	SYN	52	4	82	R
23	ETB	53	5	83	S
24	CAN	54	6	84	T
25	EM	55	7	85	U
26	SUB	56	8	86	V
27	Esc	57	9	87	W
28	FS	58	:	88	X
29	GS	59	;	89	Y

Table 6-1. Standard Set of Characters (continued)

Number	Character	Number	Character	Number	Character
90	Z	120	x	150	û
91	[	121	y	151	ù
92	\	122	z	152	ÿ
93	]	123	{	153	Ö
94	^	124	|	154	Ü
95	_	125	}	155	¢
96	‘	126	~	156	£
97	a	127	▓	157	¥
98	b	128	Ç	158	₧
99	c	129	ü	159	ƒ
100	d	130	é	160	á
101	e	131	â	161	í
102	f	132	ä	162	ó
103	g	133	à	163	ú
104	h	134	å	164	ñ
105	i	135	ç	165	Ñ
106	j	136	ê	166	ª
107	k	137	ë	167	º
108	l	138	è	168	¿
109	m	139	ï	169	⌐
110	n	140	î	170	¬
111	o	141	ì	171	½
112	p	142	Ä	172	¼
113	q	143	Å	173	¡
114	r	144	É	174	«
115	s	145	æ	175	»
116	t	146	Æ	176	░
117	u	147	ô	177	▒
118	v	148	ö	178	▓
119	w	149	ò	179	│

Table 6-1. Standard Set of Characters (continued)

Number	Character	Number	Character	Number	Character
180	┤	210	╥	240	≡
181	╡	211	╙	241	±
182	╢	212	╘	242	≥
183	╖	213	╒	243	≤
184	╕	214	╓	244	⌠
185	╣	215	╫	245	⌡
186	║	216	╪	246	÷
187	╗	217	┘	247	≈
188	╝	218	┌	248	°
189	╜	219	█	249	•
190	╛	220	▄	250	·
191	┐	221	▌	251	√
192	└	222	▐	252	ⁿ
193	┴	223	▀	253	²
194	┬	224	α	254	■
195	├	225	β	255	Space
196	─	226	Γ		
197	┼	227	π		
198	╞	228	Σ		
199	╟	229	σ		
200	╚	230	μ		
201	╔	231	τ		
202	╩	232	Φ		
203	╦	233	Θ		
204	╠	234	Ω		
205	═	235	δ		
206	╬	236	∞		
207	╧	237	ϕ		
208	╨	238	ϵ		
209	╤	239	∩		

```
A> COPY  CON  FORMAT.NTS
```

The next line is easier. Use 179 for the vertical bars, as shown here:

```
A> COPY  CON  FORMAT.NTS

WARNING!
```

The third line uses 192, 196, and 217. Completing your file results in

```
A> COPY  CON  FORMAT.NTS

WARNING!

FORMAT destroys the contents of your target diskette!
^Z
            1 File(s) copied

A>
```

Use the DOS TYPE command to display your file.

```
A> TYPE  FORMAT.NTS

WARNING!

FORMAT destroys the contents of your target diskette!

A>
```

ALT will be used in this manner in later chapters of this book.

Tips and Traps Summary

Restarting DOS with CTRL-ALT-DEL

Trap: If you restart DOS while a program is running, you risk losing all the files that the program was using.

Trap: DOS must be present in drive A, or drive A must be open (on fixed-disk systems) in order for DOS to restart.

Deleting Command-Line Characters with BACKSPACE

Tip: The BACKSPACE key directs DOS to delete the character that immediately precedes the cursor.

Canceling a Command Line with ESC

Tip: The ESC key directs DOS to ignore the command line that precedes it.

Terminating a Program with CTRL-BREAK

Tip: In many cases, the CTRL-C key combination allows you to terminate the current command.

Trap: If you use CTRL-C to terminate a program that has open files, you may lose the contents of those files.

Tip: You can use the key combinations CTRL-C and CTRL- BREAK interchangeably to terminate a program.

Trap: CTRL-C cannot terminate all programs.

Tips and Traps Summary (*continued*)

Suspending Output with CTRL-S

Tip: You can suspend the output of a command temporarily by pressing the CTRL-S key combination. To resume output, press any key.

Printing the Contents of Your Current Screen with SHIFT-PRTSC

Tip: The SHIFT-PRTSC key combination directs DOS to print the contents of the current screen.

Trap: By default, DOS uses LPT1 as its printer port. Serial printer users must direct DOS to use COM1.

Tip: Since, by default, DOS uses the LPT1 port for your printer output, the DOS MODE command allows you to direct DOS to use COM1.

Trap: By default, SHIFT-PRTSC prints only text, not graphics.

Tip: The DOS GRAPHICS command allows SHIFT-PRTSC to print screen contents that contain graphics images.

Tip: You execute the DOS GRAPHICS command only one time per session.

Trap: The printout of your graphics image may be too dark.

Tip: The /R qualifier directs the GRAPHICS command to print dark screen images as white and white screen images as dark.

Tips and Traps Summary (*continued*)

Echoing Text to Your Printer with CTRL-PRTSC

Tip: The CTRL-PRTSC key combination directs DOS to send the characters that it writes to your screen display, to your printer as well.

Tip: The CTRL-PRTSC key combination works as a toggle. The first time you press it, DOS enables character echo to your printer. The second time you press it, DOS disables character echo.

Using the DOS Function Keys

Tip: Each time you issue a command, DOS stores that command in a buffer, which allows you to to repeat or modify the command.

Tip: The F1 function key copies a single character from the command buffer for use in your current command line.

Tip: The F3 function key copies the entire command buffer to your command line.

Trap: DOS keeps track of the number of characters in your current command line and copies only the characters past that point from the command buffer.

Editing Text in the Command Buffer

Tip: The INS key allows you to insert one or more characters into the previous command line. INS works as a toggle: the first press enables insert mode; the second press enables overstrike mode.

Tip: You can use the DEL key to delete one or more characters from the previous command line.

Tips and Traps Summary (*continued*)

Tip: The F2 function key copies all of the characters up to but not including the character that you type immediately after pressing F2.

Tip: The F4 function key directs DOS to delete all of the characters in the previous command line preceding the character that you type immediately after pressing F4.

Tip: The F5 function key allows you to edit your current command line.

7

Redirecting DOS Input and Output

All of the commands examined in previous chapters have written their output to the screen display and obtained their input from the keyboard, as shown in Figure 7-1. As you will see in this chapter, DOS allows you to change (redirect) the destination of a program's output and the source of a program's input.

Since you are changing (redirecting) a program's input and output locations, this process is called *I/O redirection.* By placing unique operators in your command line, you instruct DOS to perform I/O redirection. This chapter examines each of the I/O redirection operators in detail.

Redirecting Command Output with >

At the DOS prompt, issue the DIR command

```
A> DIR
```

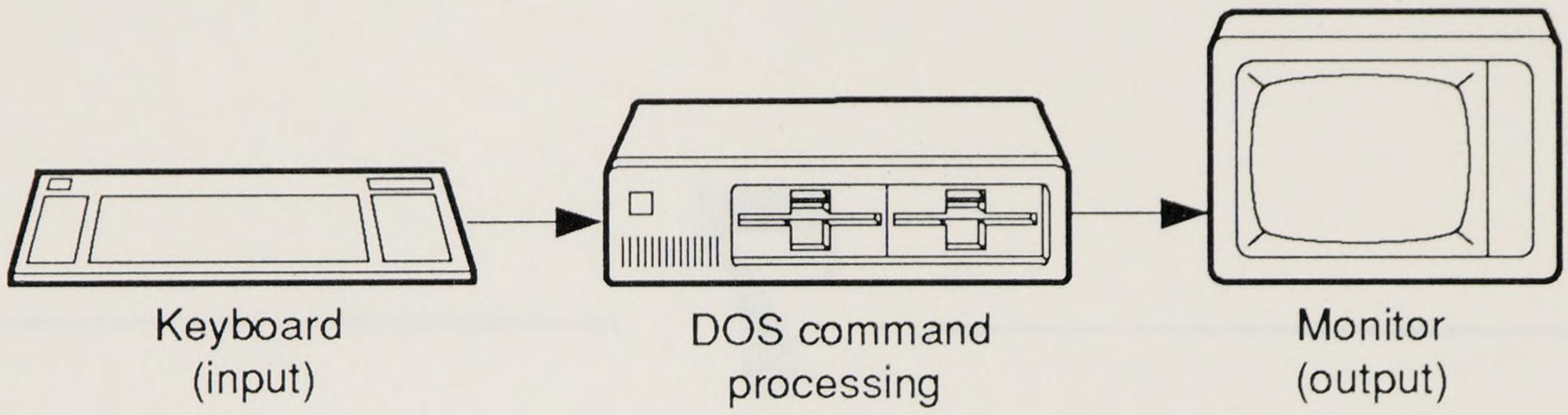


Figure 7-1. Input from keyboard and output to screen display

DOS displays the directory listing on your screen. The command uses the default input and output (see Figure 7-2).

Next, turn on your printer and issue either the command

```
A> DIR > PRN
```

or the command

```
A> DIR > COM1
```

depending on whether you have a serial or parallel printer. In this case, DOS does not display the directory listing on your screen. Instead, it sends the output of the DIR command to your printer (see Figure 7-3). The greater-than symbol, >, is the DOS *output redirection operator.*

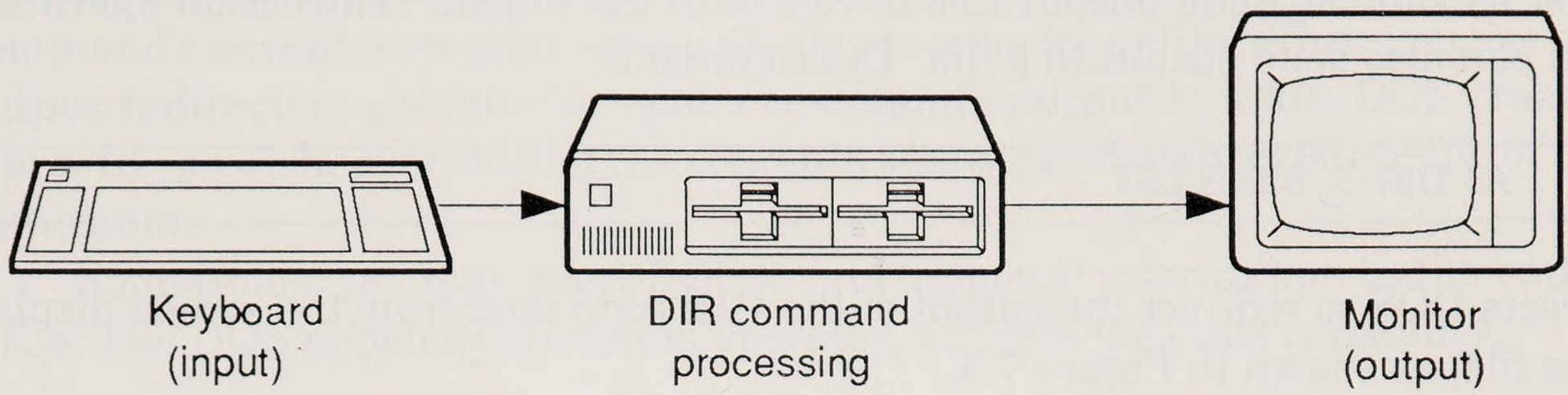


Figure 7-2. DIR command using default input and output

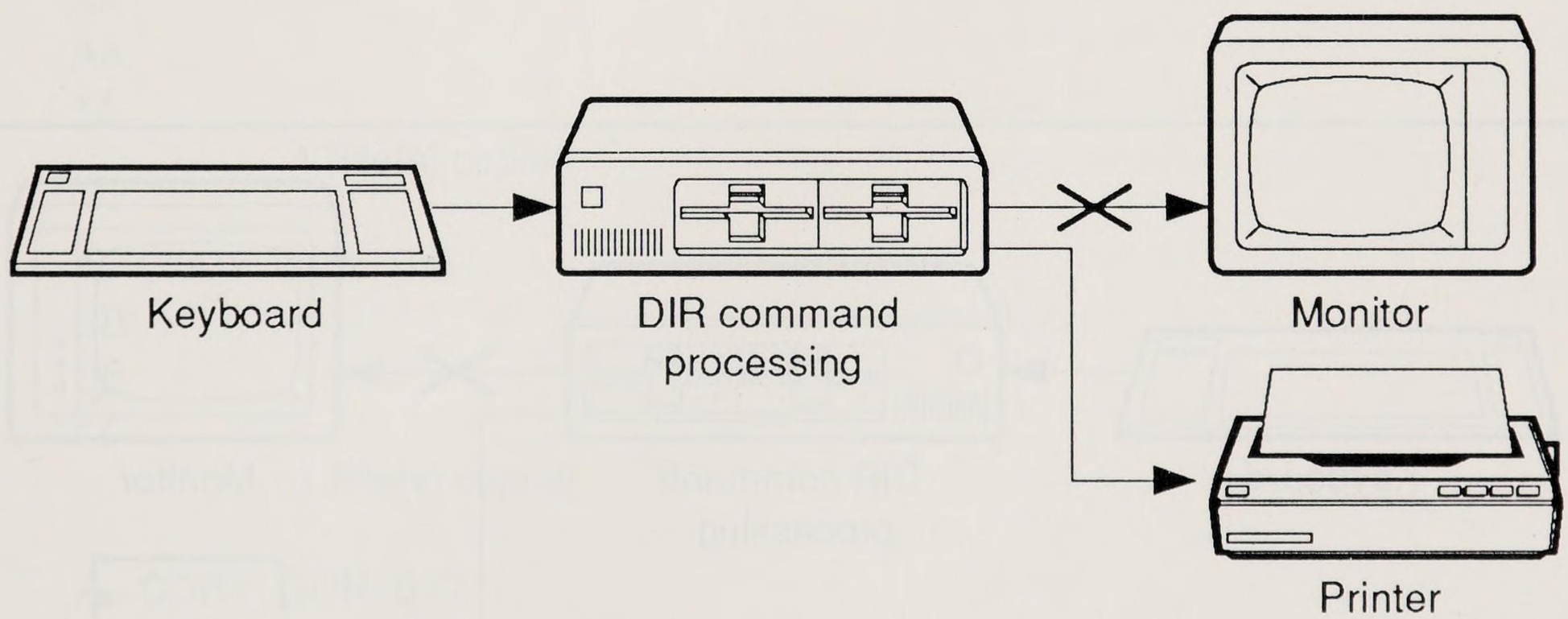


Figure 7-3. Output of DIR command being sent to printer

Tip: *The greater-than symbol, >, instructs DOS to redirect a command's output from the screen display to a device or file.*

Just as you can send output to a device with the output redirection operator, you can also send output to a file. The command

```
A> DIR > B:DIR.LST
```

directs DOS to redirect the output of the DIR command from the screen display to a file, as shown in Figure 7-4.

Trap: *If you redirect output to a file name that already resides on disk, DOS will overwrite that file.*

Just as the DOS COPY command overwrites existing files that have the same name as the target, so too will DOS redirected output overwrite an existing file. Therefore, be careful to ensure that your I/O redirection does not overwrite a needed file. Meaningful file names are your best protection against errant DOS commands.

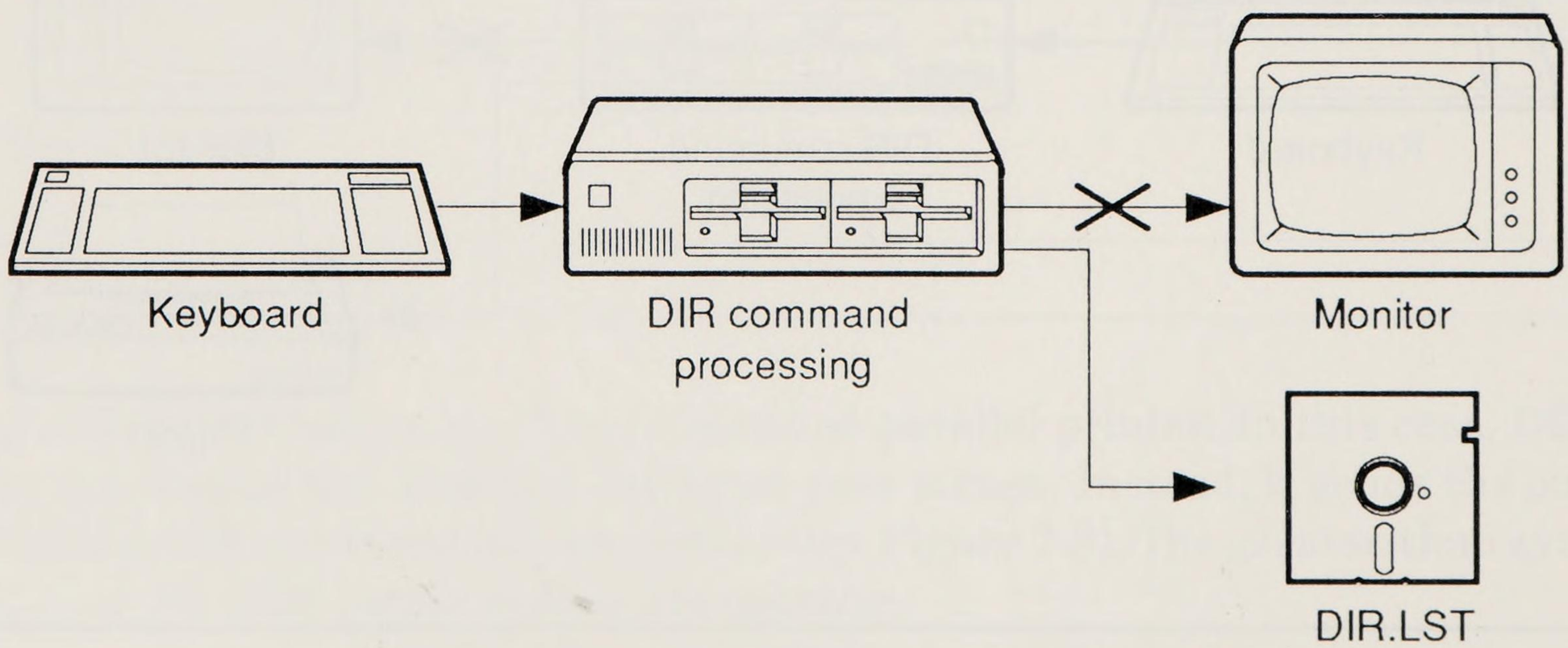


Figure 7-4. Output of DIR sent to file

Appending Redirected Output with >>

As you just saw, the DOS output redirection operator allows you to redirect a command's output from your screen display to a device or file. When you use the output redirection operator to send a command's output to a file, DOS creates a new file on disk, overwriting the contents of any previously existing file of the same name.

Sometimes, you may want DOS to add (append) information to the end of a file. The DOS append redirection operator, >>, provides this capability.

Tip: *The DOS append redirection operator, >>, instructs DOS to send the output of a command to a file or device. If a file matching the name specified in the command line already exists on disk, DOS will append the redirected output to the file.*

So that you have several files to experiment with, create the following:

```
A> COPY  CON  B:A
AA
AA
^Z
            1 File(s) copied

A> COPY  CON  B:B
BB
BB
^Z
            1 File(s) copied

A> COPY  CON  B:C
CC
CC
^Z
            1 File(s) copied
```

Next, issue the command

```
A> TYPE  B:A
AA
AA

A>
```

As you can see, if you omit the DOS I/O redirection operators, DOS displays your command output on the screen display as before. However, if you issue the command

```
A> TYPE  B:A >> B:LETTERS.DAT
A>
```

DOS directs the output of the TYPE command from the screen to the file on drive B named LETTERS.DAT.

Tip: *If the target file of an append redirection operation does not exist on disk, DOS will create it.*

In this example, because the LETTERS file did not exist on the disk in drive B, DOS created the file.

```
A> TYPE  B:LETTERS.DAT
AA
AA

A>
```

Now use the append redirection operator as shown here:

```
A> TYPE  B:B >> B:LETTERS.DAT
A>
```

As before, DOS redirects the output from the screen display, appending it to the LETTERS.DAT file on drive B.

```
A> TYPE  B:LETTERS.DAT
AA
AA
BB
BB

A>
```

Invoking the redirection command once more

```
A> TYPE  B:C  >>  B:LETTERS.DAT
```

results in the file

```
A> TYPE  B:LETTERS.DAT
AA
AA
BB
BB
CC
CC

A>
```

In each case, the append redirection operator, >>, instructed DOS to append the output of the command to the LETTERS.DAT file.

Redirecting Command Input with <

DOS also provides an input redirection operator. The less-than symbol, <, instructs DOS to redirect a command's source of input from the keyboard to a file.

Tip: *The DOS input redirection operator, <, instructs DOS to redirect a command's input from the keyboard to a file.*

Before examining the DOS input redirection operator, consider some of the DOS commands that specifically support DOS I/O redirection. With your Program Disk in drive A, issue the command

```
A> DIR  MORE.COM

Volume in drive A is MS330PP01
Directory of  A:\

MORE        COM          282        7-24-87  12:00a
            1 File(s) 4096 bytes free
```

Tip: *MORE is an external DOS command that allows you to display output a screenful at a time.*

Next, issue the command

```
A> DIR  SORT.EXE

Volume in drive A is MS330PP01
Directory of  A:\

SORT        EXE          1946          7-24-87  12:00a
            1 File(s) 4096 bytes free
```

Tip: *SORT is an external DOS command that allows you to display output sorted in either ascending or descending order.*

Finally, issue the command

```
A> DIR  FIND.EXE

Volume in drive A is MS330PP01
Directory of  A:\

FIND        EXE          6403          7-24-87  12:00a
            1 File(s) 4096 bytes free
```

Tip: *FIND is an external DOS command that allows you to display each line of output containing a specific word or phrase.*

These commands will be used throughout the remainder of the chapter. Keep in mind that each of these commands is an external DOS command, so your Program Disk must be available.

With your Program Disk in drive A, issue the command

```
A> DIR > B:DIR.LST
```

DOS will create a file on drive B that contains the directory listing of your Program Disk.

```
A> DIR  B:DIR.LST

Volume in drive B is MISC
Directory of  B:\

DIR         LST                1534          6-22-88  11:29a
            1 File(s) 253760 bytes free
```

Next, use the MORE command, as shown here, to display the contents of the DIR.LST file a screenful at a time.

```
A> MORE < B:DIR.LST
```

```
A> MORE  B:DIR.LST

Volume in drive A is MS330PP01
Directory of  A:\

4201          CPI        17089   7-24-87  12:00a
5202          CPI        459     7-24-87  12:00a
ANSI          SYS        1647    7-24-87  12:00a
APPEND        EXE        5794    7-24-87  12:00a
ASSIGN        COM        1530    7-24-87  12:00a
ATTRIB        EXE        10656   7-24-87  12:00a
CHKDSK        COM        9819    7-24-87  12:00a
CONFIG        SYS        10      6-05-88   5:50p
COMP          COM        4183    7-24-87  12:00a
COUNTRY       SYS        11254   7-24-87  12:00a
DISKCOMP      COM        5848    7-24-87  12:00a
DISKCOPY      COM        6264    7-24-87  12:00a
DISPLAY       SYS        11259   7-24-87  12:00a
DRIVER        SYS        1165    7-24-87  12:00a
EDLIN         COM        7495    7-24-87  12:00a
EXE2BIN       EXE        3050    7-24-87  12:00a
FASTOPEN      EXE        3888    7-24-87  12:00a
EGA           CPI        49065   7-24-87  12:00a
FIND          EXE        6403    7-24-87  12:00a
FORMAT        COM        11671   7-24-87  12:00a

-- MORE --
```

Figure 7-5. Output suspended by MORE command

If you displayed the contents of DIR.LST with the TYPE command instead, the information would scroll past rapidly on the screen. The MORE command will display a screenful of information, pause, and prompt you with

```
--MORE--
```

(see Figure 7-5). Press any key to continue. MORE will display the next screenful of information (if one exists) or stop at the end of the file (see Figure 7-6).

```
4201        CPI      17089   7-24-87  12:00a
5202        CPI      459     7-24-87  12:00a
ANSI        SYS      1647    7-24-87  12:00a
APPEND      EXE      5794    7-24-87  12:00a
ASSIGN      COM      1530    7-24-87  12:00a
ATTRIB      EXE      10656   7-24-87  12:00a
CHKDSK      COM      9819    7-24-87  12:00a
CONFIG      SYS      10      6-05-88   5:50p
COMP        COM      4183    7-24-87  12:00a
COUNTRY     SYS      11254   7-24-87  12:00a
DISKCOMP    COM      5848    7-24-87  12:00a
DISKCOPY    COM      6264    7-24-87  12:00a
DISPLAY     SYS      11259   7-24-87  12:00a
DRIVER      SYS      1165    7-24-87  12:00a
EDLIN       COM      7495    7-24-87  12:00a
EXE2BIN     EXE      3050    7-24-87  12:00a
FASTOPEN    EXE      3888    7-24-87  12:00a
EGA         CPI      49065   7-24-87  12:00a
FIND        EXE      6403    7-24-87  12:00a
FORMAT      COM      11671   7-24-87  12:00a

-- MORE --

GRAFTABL    COM      6136    7-24-87  12:00a
GRAPHICS    COM      13943   7-24-87  12:00a
JOIN        EXE      9612    7-24-87  12:00a
KEYB        COM      9041    7-24-87  12:00a
LABEL       COM      2346    7-24-87  12:00a
MODE        COM      15440   7-24-87  12:00a
MORE        COM      282     7-24-87  12:00a
NLSFUNC     EXE      3029    7-24-87  12:00a
PRINT       COM      8995    7-24-87  12:00a
RECOVER     COM      4268    7-24-87  12:00a
SELECT      COM      4132    7-24-87  12:00a
SORT        EXE      1946    7-24-87  12:00a
SUBST       EXE      10552   7-24-87  12:00a
SYS         COM      4725    7-24-87  12:00a
COMMAND     COM      25276   7-24-87  12:00a
          35 File(s)     4096 bytes free
```

Figure 7-6. Resumption of output after pressing any key

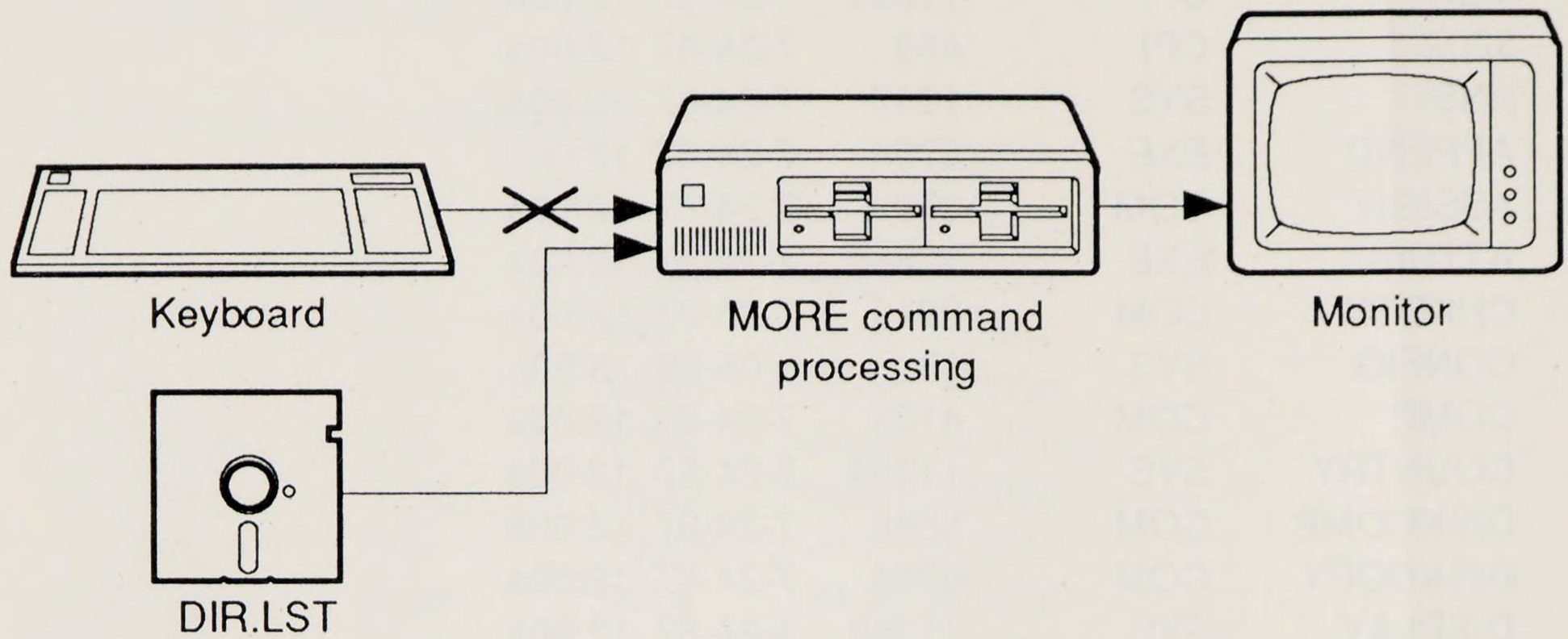


Figure 7-7. Redirecting output of MORE command

In this example, you have redirected MORE's input source from the keyboard to a file, as shown in Figure 7-7.

Create the NUMBERS.DAT file, as shown here:

```
A> COPY CON B:NUMBERS.DAT
3
1
4
2
7
5
6
^Z
        1 File(s) copied

A>
```

Next, use the DOS SORT command in the following format:

```
A> SORT < B:NUMBERS.DAT
1
2
```

```
3
4
5
6
7

A>
```

You have now redirected the input of the command from the keyboard to a file.

In a similar manner, you can instruct SORT to sort the information in highest to lowest order by using the /R qualifier, as shown here:

```
A> SORT /R < B:NUMBERS.DAT
7
6
5
4
3
2
1

A>
```

Tip: *The /R qualifier instructs SORT to sort values in reverse order, from highest to lowest.*

Likewise, if your file contains several columns of information, as in

UCLA	CA	Bruins
USC	CA	Trojans
UW	WA	Huskies
WSU	WA	Cougars
UNLV	NV	Rebels
ASU	AZ	Sun Devils
UA	AZ	Wildcats

SORT allows you to sort the information on a specific column.

Tip: *The SORT /+n qualifier allows you to direct SORT to sort input based on characters in column n. To sort data on the character in column 5, for example, you would use /+5.*

Given the file

```
A> TYPE  TEAMS.DAT
UCLA      CA      Bruins
USC       CA      Trojans
UW        WA      Huskies
WSU       WA      Cougars
UNLV      NV      Rebels
ASU       AZ      Sun Devils
UA        AZ      Wildcats

A>
```

you can sort teams by state using

```
A> SORT /+14 < TEAMS.DAT
ASU       AZ      Sun Devils
UA        AZ      Wildcats
UCLA      CA      Bruins
USC       CA      Trojans
UNLV      NV      Rebels
WSU       WA      Cougars
UW        WA      Huskies

A>
```

The DOS FIND command allows you to locate each occurrence of a word or phrase. To display the lines in the DIR.LST file that contain the letters EXE, enter

```
A> FIND "EXE" < B:DIR.LST
APPEND      EXE      5794    7-24-87  12:00a
ATTRIB      EXE      10656   7-24-87  12:00a
EXE2BIN     EXE      3050    7-24-87  12:00a
FASTOPEN    EXE      3888    7-24-87  12:00a
FIND        EXE      6403    7-24-87  12:00a
JOIN        EXE      9612    7-24-87  12:00a
NLSFUNC     EXE      3029    7-24-87  12:00a
SORT        EXE      1946    7-24-87  12:00a
SUBST       EXE      10552   7-24-87  12:00a

A>
```

If you want to precede each line containing the desired word or phrase with its corresponding line number, use the /N qualifier in the FIND command.

Tip: *The /N qualifier instructs the FIND command to precede each line displayed with its line number.*

The command

```
A> FIND /N "EXE" < B:DIR.LST
```

will display

```
A> FIND /N "EXE" < B:DIR.LST
[8]APPEND      EXE      5794    7-24-87  12:00a
[10]ATTRIB     EXE      10656   7-24-87  12:00a
[20]EXE2BIN    EXE      3050    7-24-87  12:00a
[21]FASTOPEN   EXE      3888    7-24-87  12:00a
[23]FIND       EXE      6403    7-24-87  12:00a
[27]JOIN       EXE      9612    7-24-87  12:00a
[32]NLSFUNC    EXE      3029    7-24-87  12:00a
[36]SORT       EXE      1946    7-24-87  12:00a
[37]SUBST      EXE      10552   7-24-87  12:00a

A>
```

FIND also allows you to display the lines that do not contain the word or phrase you specify.

Tip: *The /V qualifier instructs the FIND command to display lines of the file that do not contain the specified word or phrase.*

For example, the command

```
A> FIND /V "EXE" < B:DIR.LST
```

will display all the lines in the DIR.LST file that do not contain the letters "EXE."

Finally, if you want to count the number of times that a word or phrase occurs in a file, use the FIND /C qualifier.

Tip: *The /C qualifier instructs the FIND command to display a count of the number of times the specified word or phrase appears.*

Given the TEAMS.DAT file, which contains

```
A> TYPE  TEAMS.DAT
UCLA      CA      Bruins
USC       CA      Trojans
UW        WA      Huskies
WSU       WA      Cougars
UNLV      NV      Rebels
ASU       AZ      Sun Devils
UA        AZ      Wildcats

A>
```

the command

```
A> FIND /C "AZ" < B:TEAMS.DAT
```

will display

```
A> FIND /C "AZ" < B:TEAMS.DAT
2

A>
```

Trap: *The FIND /C qualifier overrides /V and /N.*

The FIND /C qualifier instructs FIND to display a count of the number of occurrences of the input word or phrase. If you issue the command

```
A> FIND /N /C "AZ" < B:TEAMS.DAT
```

the /C qualifier will override /N and FIND will display

```
A> FIND /N /C "AZ" < B:TEAMS.DAT
2

A>
```

Failing to Redirect MORE and SORT

As you have just seen, the DOS MORE and SORT commands expect redirected input and output. If you issue either command from the DOS prompt without redirection, as in

```
A> SORT
```

DOS will wait for you to type information at the keyboard.
Type the numbers that follow:

```
A> SORT
3
1
4
2
5
```

To inform SORT that you have entered all of the data, press F6 and ENTER. SORT will display your sorted data, as shown here:

```
A> SORT
3
1
4
2
5
^Z
1
2
3
4
5

A>
```

In this case, the command has not been redirected, leaving its input source and output destination the keyboard and screen, as shown in Figure 7-8.

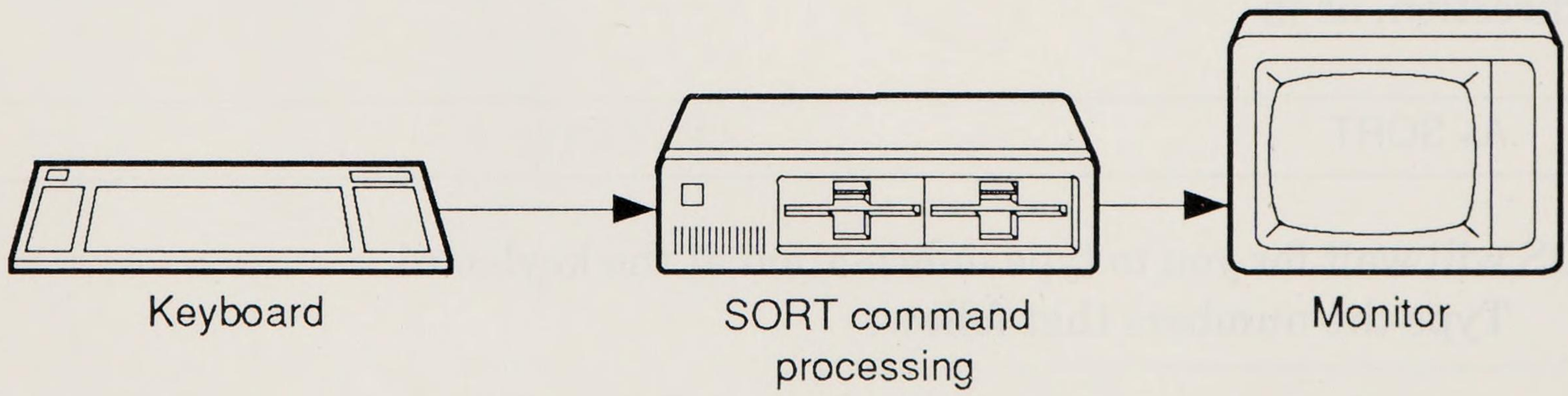


Figure 7-8. Keyboard as input source and screen as destination of output

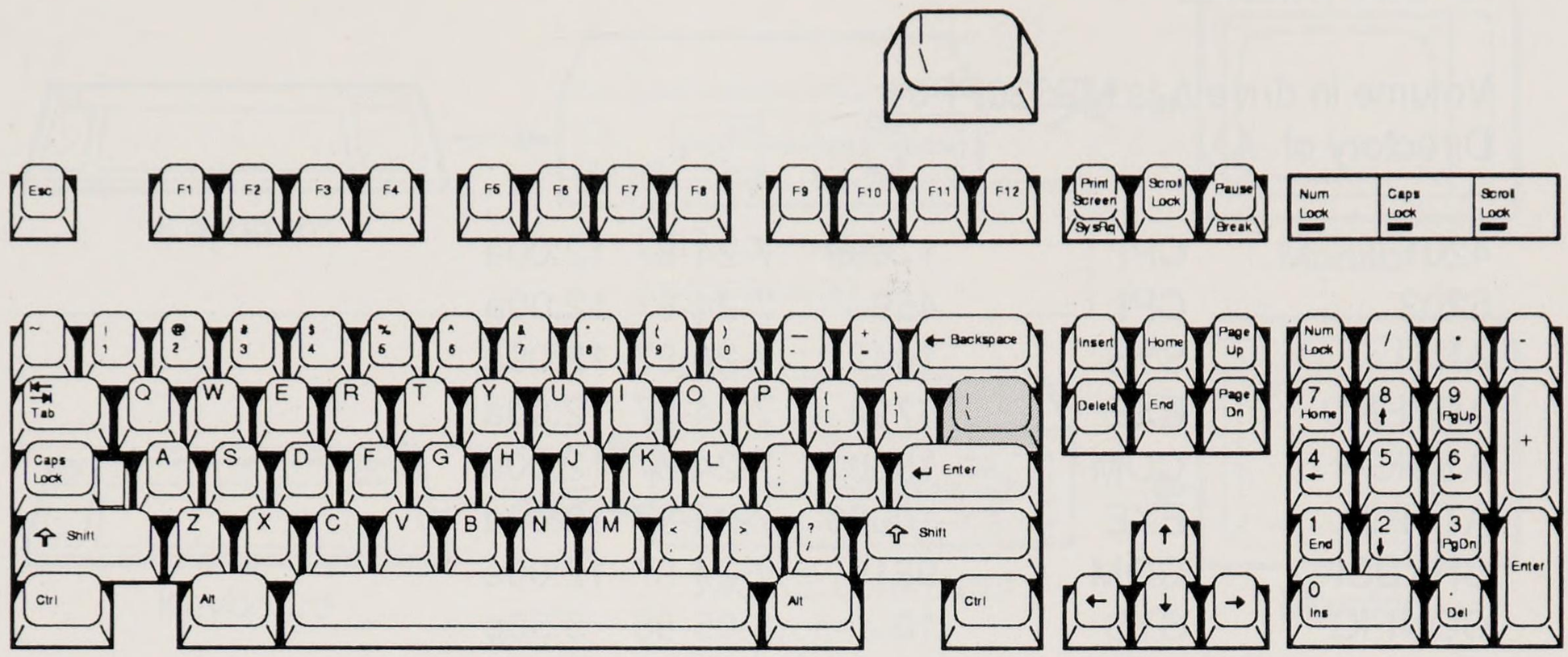


Figure 7-9. Location of pipe character

The DOS MORE command works in an identical manner if you do not include redirection when you invoke it.

```
A> MORE
```

Using the DOS Pipe Operator (|)

The final DOS I/O redirection operator examined here is the DOS pipe (|). Locate the key used to represent the pipe operator (Figure 7-9).

Unlike the other DOS I/O redirection operators, which allow you to redirect input or output to or from files, the DOS pipe allows you to make the output of one program become the input to a second program.

Tip: *The DOS pipe operator, |, directs DOS to make the output of one program the input of a second program.*

```
A> DIR | MORE

Volume in drive A is MS330PP01
Directory of  A:\

4201          CPI        17089   7-24-87  12:00a
5202          CPI        459     7-24-87  12:00a
ANSI          SYS        1647    7-24-87  12:00a
APPEND        EXE        5794    7-24-87  12:00a
ASSIGN        COM        1530    7-24-87  12:00a
ATTRIB        EXE        10656   7-24-87  12:00a
CHKDSK        COM        9819    7-24-87  12:00a
CONFIG        SYS        10      6-05-88   5:50p
COMP          COM        4183    7-24-87  12:00a
COUNTRY       SYS        11254   7-24-87  12:00a
DISKCOMP      COM        5848    7-24-87  12:00a
DISKCOPY      COM        6264    7-24-87  12:00a
DISPLAY       SYS        11259   7-24-87  12:00a
DRIVER        SYS        1165    7-24-87  12:00a
EDLIN         COM        7495    7-24-87  12:00a
EXE2BIN       EXE        3050    7-24-87  12:00a
FASTOPEN      EXE        3888    7-24-87  12:00a
EGA           CPI        49065   7-24-87  12:00a
FIND          EXE        6403    7-24-87  12:00a
FORMAT        COM        11671   7-24-87  12:00a

-- MORE --
```

Figure 7-10. Use of pipe with MORE command

Before experimenting with the DOS pipe operator, remove the write-protect tab from your Program Disk and insert the disk in drive A. Issue the DIR command shown in Figure 7-10. As you can see, DOS has redirected the output of the DIR command to become the input of the MORE command. This process is illustrated in Figure 7-11.

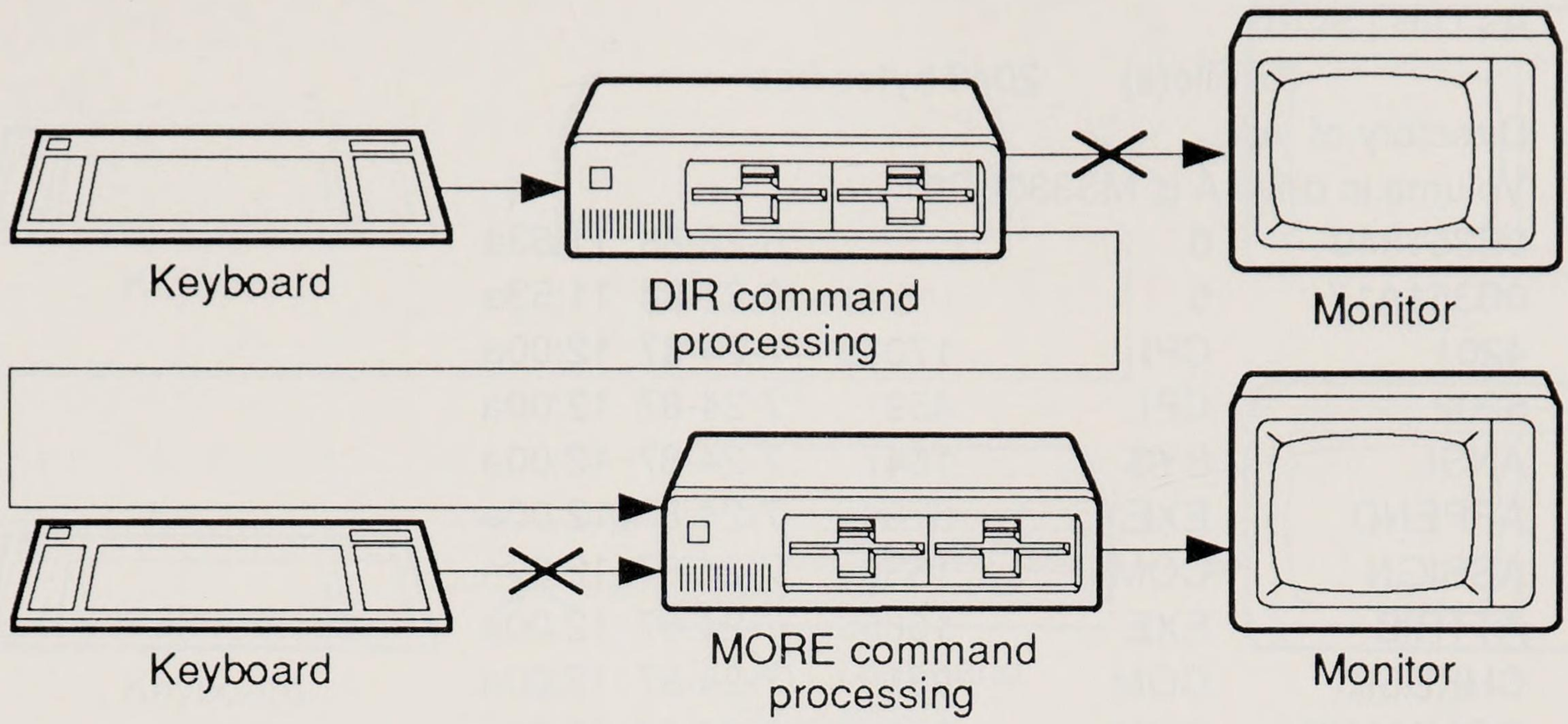


Figure 7-11. Redirected output with pipe

Similarly, you can obtain a sorted directory listing with the command

```
A> DIR | SORT
```

The command produces the sorted directory listing shown in Figure 7-12. In this case, the I/O redirection becomes that shown in Figure 7-13.

Trap: *DOS cannot use the pipe operator if the default disk is write protected.*

When you use the DOS pipe in your command line, DOS creates two temporary files to assist in its processing. If you examine the sorted directory listing in Figure 7-12, for example, you can identify these two files:

0B351840	0	6-22-88	11:53a
0B351A17	0	6-22-88	11:53a

```
A> DIR | SORT
            37 File(s)     2048 bytes free
Directory of  A:\
Volume in drive A is MS330PP01
0B351840        0                    6-22-88  11:53a
0B351A17        0                    6-22-88  11:53a
4201            CPI          17089   7-24-87  12:00a
5202            CPI          459     7-24-87  12:00a
ANSI            SYS          1647    7-24-87  12:00a
APPEND          EXE          5794    7-24-87  12:00a
ASSIGN          COM          1530    7-24-87  12:00a
ATTRIB          EXE          10656   7-24-87  12:00a
CHKDSK          COM          9819    7-24-87  12:00a
COMMAND         COM          25276   7-24-87  12:00a
COMP            COM          4183    7-24-87  12:00a
CONFIG          SYS          10      6-05-88   5:50p
COUNTRY         SYS          11254   7-24-87  12:00a
DISKCOMP        COM          5848    7-24-87  12:00a
DISKCOPY        COM          6264    7-24-87  12:00a
DISPLAY         SYS          11259   7-24-87  12:00a
DRIVER          SYS          1165    7-24-87  12:00a
EDLIN           COM          7495    7-24-87  12:00a
EGA             CPI          49065   7-24-87  12:00a
EXE2BIN         EXE          3050    7-24-87  12:00a
FASTOPEN        EXE          3888    7-24-87  12:00a
FIND            EXE          6403    7-24-87  12:00a
FORMAT          COM          11671   7-24-87  12:00a
GRAFTABL        COM          6136    7-24-87  12:00a
GRAPHICS        COM          13943   7-24-87  12:00a
JOIN            EXE          9612    7-24-87  12:00a
KEYB            COM          9041    7-24-87  12:00a
LABEL           COM          2346    7-24-87  12:00a
MODE            COM          15440   7-24-87  12:00a
MORE            COM          282     7-24-87  12:00a
NLSFUNC         EXE          3029    7-24-87  12:00a
PRINT           COM          8995    7-24-87  12:00a
RECOVER         COM          4268    7-24-87  12:00a
SELECT          COM          4132    7-24-87  12:00a
SORT            EXE          1946    7-24-87  12:00a
SUBST           EXE          10552   7-24-87  12:00a
SYS             COM          4725    7-24-87  12:00a
```

Figure 7-12. Sorted directory listing

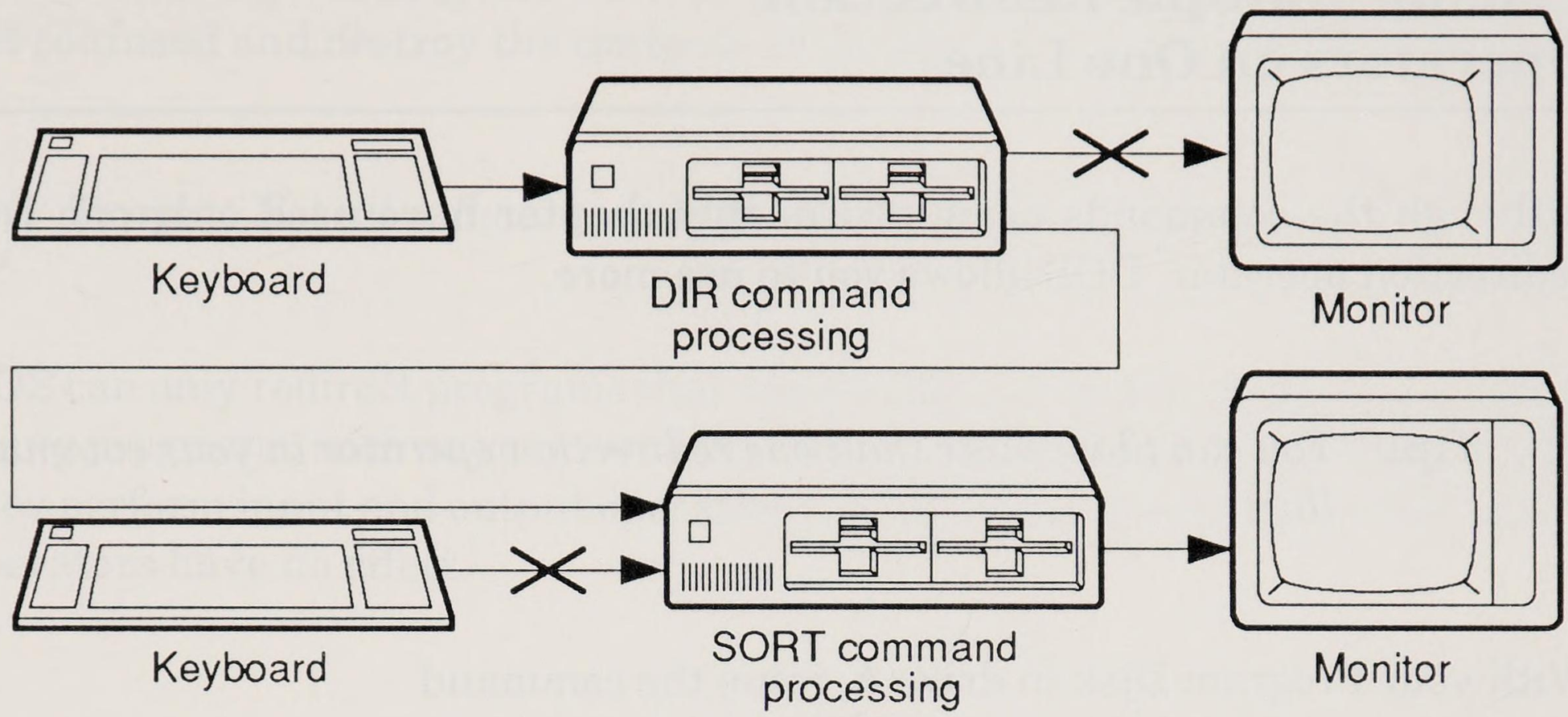


Figure 7-13. Redirected output with SORT

When the pipe command completes, DOS deletes these two files. If your default disk is write protected, a command such as

```
A> DIR | SORT
```

fails, because DOS cannot create the temporary files. In this case, DOS will display the message

```
Write protect error writing drive A
Abort, Retry, Fail?
```

If this occurs, simply remove the write-protect tab from your disk and repeat the command. In a similar manner, if your current disk does not contain enough free space for the temporary files, the command will fail.

Trap: *If DOS displays the message*

```
Intermediate file error during pipe
```

during a pipe operation, your disk does not have enough space to complete the operation.

Placing Multiple Redirection Operators on One Line

Although the commands examined in this chapter have used only one DOS redirection operator, DOS allows you to use more.

Tip: *You can place more than one redirection operator in your command line.*

With your Program Disk in drive A, issue the command

```
A> DIR | SORT | MORE
```

DOS will display a sorted directory listing one screenful at a time. This redirection is illustrated in Figure 7-13.

To display the contents of the DIR.LST file a screenful at a time, sorted by state, you have two choices. First, you can use the DOS pipe operator:

```
A> TYPE B:DIR.LST | SORT | MORE
```

Second, you can combine different I/O redirection operators:

```
A> SORT < B:DIR.LST | MORE
```

These two commands are identical in function.

Trap: *If you use the same file name for input and output on opposing sides of a redirection operator, you are likely to lose the contents of your file.*

With the same file name on opposite sides of a redirection operator, as in

```
A> SORT < FILENAME.EXT > FILENAME.EXT
```

DOS is actually reading and writing to the same file. It is likely that DOS will get confused and destroy the contents of the file.

Trap: *Not all software packages support I/O redirection.*

DOS can only redirect programs that specifically use DOS for input and output. Many programs, such as your word processor or spreadsheet, bypass DOS when they perform input and output operations. With these programs, the redirection operators have no effect.

Tips and Traps Summary

Redirecting Command Output with >

Tip: The greater-than symbol, >, instructs DOS to redirect a command's output from the screen display to a device or file.

Trap: If you redirect output to a file name that already resides on disk, DOS will overwrite that file.

Appending Redirected Output with >>

Tip: The DOS append redirection operator, >>, instructs DOS to send the output of a command to a file or device. If a file matching the name specified in the command line already exists on disk, DOS will append the redirected output to the file.

Tip: If the target file of an append redirection operation does not exist on disk, DOS will create it.

Redirecting Command Input with <

Tip: The DOS input redirection operator, <, instructs DOS to redirect a command's input from the keyboard to a file.

Tip: MORE is an external DOS command that allows you to display output a screenful at a time.

Tip: SORT is an external DOS command that allows you to display output sorted in either ascending or descending order.

Tip: FIND is an external DOS command that allows you to display each line of output containing a specific word or phrase.

Tips and Traps Summary (*continued*)

Tip: The SORT /+*n* qualifier allows you to direct SORT to sort input based on column *n*. To sort data on the character in column 5, for example, you would use /+5.

Tip: The /N qualifier instructs the FIND command to precede each line displayed with its line number.

Tip: The /V qualifier instructs the FIND command to display lines of the file that do not contain the specified word or phrase.

Tip: The /C qualifier instructs the FIND command to display a count of the number of times the specified word or phrase appears.

Trap: The FIND /C qualifier overrides /V and /N.

Using the DOS Pipe Operator (|)

Tip: The DOS pipe operator, |, directs DOS to make the output of one program the input of a second program.

Trap: DOS cannot use the pipe operator if the default disk is write protected.

Trap: If DOS displays the message

```
Intermediate file error during pipe
```

during a pipe operation, your disk does not have enough space to complete the operation.

Placing Multiple Redirection Operators on One Line

Tip: You can place more than one redirection operator in your command line.

Tips and Traps Summary (*continued*)

Trap: If you use the same file name for input and output on opposing sides of a redirection operator, you are likely to lose the contents of your file.

Trap: Not all software packages support I/O redirection.

8

Using the DOS PRINT Command

Throughout this book you have used several different techniques to obtain printed (or hardcopy) listings of your files:

- The SHIFT-PRTSC key combination to print the current contents of your screen display
- The CTRL-PRTSC key combination to enable character echoing to your printer
- The COPY FILENAME.EXT LPT1 command to copy files directly to your printer device

Although these methods allow you to obtain your desired printed output, DOS provides a much more functional method: the DOS PRINT command. This chapter will examine the DOS PRINT command in detail, experimenting with each of the qualifiers that you can place in the PRINT command line. PRINT is a very powerful command with several convenient features. Most DOS users only begin to tap PRINT's capabilities.

Getting Started with PRINT

With your Program Disk in drive A, issue the command

```
A> DIR  PRINT.COM

Volume in drive A is MS330PP01
Directory of  A:\

PRINT          COM          8995     7-24-87  12:00a
            1 File(s)     5120 bytes free
```

As you can see, PRINT is an external DOS command. In this chapter you will invoke PRINT in several different ways. To start, simply use

```
A> PRINT
```

PRINT will respond with the prompt

```
Name of list device [PRN]:
```

This prompt is PRINT's way of asking which device you want to use for your printed output. By default, PRINT uses PRN.

Tip: *By default, PRINT sends all of its output to the PRN device. If you have a parallel printer that is attached to LPT1, press ENTER at the prompt*

```
Name of list device [PRN]:
```

If you have a serial printer, type the device name ***COM1*** *and press ENTER.*

The prompt

```
Name of list device [PRN]:
```

is one of the few DOS prompts that provides a default response. In this case, the response is contained within brackets, [PRN]. If you press ENTER, PRINT will use the PRN device for all of its output.

Regardless of whether you press ENTER to select the default device PRN or type an alternative device name (such as COM1), PRINT will respond with

```
Resident part of PRINT installed
PRINT queue is empty
```

This message tells you that PRINT has placed the software it needs resident in memory. This is the software responsible for managing your printer and the files to be printed.

Tip: *If you are using a serial printer, type* ***COM1*** *at the prompt*

```
Name of list device [PRN]: COM1
```

Earlier in this book, serial printer users had to issue the command

```
A> MODE LPT1:= COM1
```

in order for print-screen and character-echo operations to send data to their serial printers. If you simply want to print files, you can specify the correct device for PRINT instead of using the DOS MODE command.

Trap: *If you specify an invalid device name at the prompt*

```
Name of list device [PRN]:
```

PRINT will not install in memory.

As just discussed, when PRINT displays the prompt

```
Name of list device [PRN]:
```

it is asking you to enter the name of the device that your printed output is to be written to. If you type an invalid device name, PRINT will display the following error message:

```
List output is not assigned to a device
```

If this occurs, reissue the PRINT command and enter the correct device name. So that you have several files to experiment with, create the following files:

```
A> COPY  CON  B:\NUMBERS.DAT
1
2
3
4
5
^Z
            1 File(s) copied

A> COPY  CON  B:LETTERS.DAT
AAA
BBB
CCC
DDD
EEE
^Z
            1 File(s) copied

A> COPY  CON  B:PRINT.NTS
PRINT is an external DOS command.
By default, PRINT uses PRN.
^Z
            1 File(s) copied

A>
```

Next, with your printer on-line and your Program Disk in drive A, issue the command

```
A> PRINT  B:NUMBERS.DAT
```

PRINT will respond with

```
A> PRINT  B:NUMBERS.DAT

     B:\NUMBERS.DAT is currently being printed

A>
```

This message tells you that PRINT is printing your file. At this time, the file should begin printing on your printer. If not, make sure that your printer is on-line.

Tip: *You can print multiple files with one PRINT command line.*

If you have multiple files to print, you can either issue several PRINT commands, printing each file one at a time, or you can place all of the file names on a single PRINT command line, as shown here:

```
A> PRINT  B:NUMBERS.DAT  B:LETTERS.DAT  B:PRINT.NTS
```

In this case, PRINT will respond with

```
B:\NUMBERS.DAT is currently being printed
B:\LETTERS.DAT is in queue
B:\PRINT.NTS is in queue
```

These messages tell you that PRINT is currently printing the NUMBERS.DAT file and that it is waiting to print the LETTERS.DAT and PRINT.NTS files.

Tip: *PRINT keeps a list of the files it needs to print. This list is called a queue.*

A *queue* is simply a waiting line. When PRINT displays the message

```
FILENAME.EXT is in queue
```

it is telling you that it has placed the file in the queue. In this case, the queue is simply a list of files waiting to be printed.

Tip: *To display the list of files in the print queue, issue the PRINT command without specifying any files.*

If you enter the PRINT command without including any file names in the command line, PRINT will list all of the files contained in the print queue, as shown here:

```
A> PRINT

B:\NUMBERS.DAT is currently being printed
B:\LETTERS.DAT is in queue
B:\PRINT.NTS is in queue

A>
```

Tip: *PRINT always prints files in the order in which they were placed in the queue.*

Assuming that your print queue contains the files

```
A> PRINT

B:\FIRST is currently being printed
B:\SECOND is in queue
B:\THIRD is in queue
B:\FOURTH is in queue

A>
```

PRINT will print the files in the exact order in which they appear, from top to bottom.

Tip: *PRINT supports the DOS wildcard characters.*

You may recall that the following command prints the LETTERS.DAT, NUMBERS.DAT, and PRINT.NTS files:

```
A> PRINT B:LETTERS.DAT B:NUMBERS.DAT B:PRINT.NTS
```

Assuming that the LETTERS.DAT and NUMBERS.DAT files are the only files on drive B with the DAT extension, you can change the previous command to read

```
A> PRINT B:*.DAT PRINT.NTS
```

In this case, PRINT will display

```
B:\NUMBERS.DAT is currently being printed
B:\LETTERS.DAT is in queue
B:\PRINT.NTS is in queue
```

Trap: *Not all files are printable.*

Just as your screen will not display nontext files, you cannot print such files. If, for example, you issue the command

```
A> PRINT COMMAND.COM
```

your printer will print several unrecognizable characters and your printer's bell will likely sound. Therefore, use PRINT only to print standard text files.

Trap: *If you delete a file currently contained in the print queue, the PRINT command will print unrecognizable characters while attempting to print a nonexistent file.*

When PRINT has placed a file in the print queue, you cannot delete the file or remove the disk containing the file, because PRINT will be unable to locate the file on disk when it is time for printing.

How PRINT Works

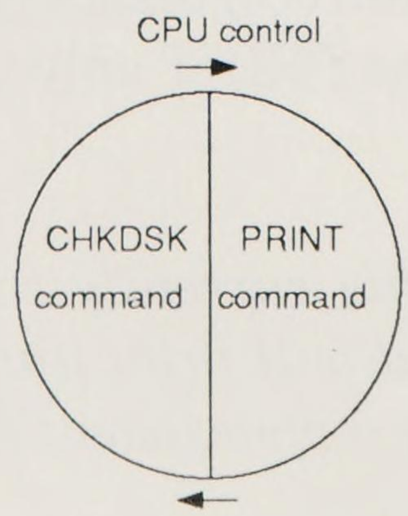
The commands examined thus far in this book have one major factor in common: each requires the command to complete before you can issue subsequent commands. PRINT, as you will see, is an exception to this rule. Printers by their mechanical nature are relatively slow devices. If you have to print a 30-page document, for example, the printing could take a considerable amount of time. If PRINT required the printing to complete before you could continue with other commands, a great deal of time would be wasted.

Instead, as soon as PRINT places a file into the print queue, it redisplays the DOS prompt so that you can continue to execute other DOS commands. This frees you to perform other tasks at the computer while your files print. Your computer appears to be performing two tasks at once.

In actuality, however, your computer can perform only one task at a time. Therefore, in order for DOS to print files while you execute other commands, your computer must quickly switch between the two tasks. Because this change occurs so fast, you normally do not realize that it is occurring. Assume, for example, that you have a file printing and that you invoke the DOS CHKDSK command:

```
A> CHKDSK
```

DOS will allow CHKDSK to perform for a period of time, then pass control to PRINT, and then to CHKDSK in a rapid *round robin* manner, as illustrated here:



This task switching maximizes your computer's potential. Later, this chapter will examine several qualifiers that allow you to improve your system performance by fine-tuning how PRINT shares the computer with other commands.

Removing Files from The Print Queue

In the same way that PRINT allows you to add files to the print queue, it also allows you to remove a file from the queue, or to terminate printing of all queued files.

Tip: *The /C qualifier instructs PRINT to remove the specified file(s) from the print queue.*

Just as you type a file name in your PRINT command line to add a file to the print queue, to remove a file from the queue you place the /C qualifier immediately following the file name, as shown here:

```
A> PRINT FILENAME.EXT /C
```

Assuming, for example, that your print queue contains

```
A> PRINT

B:\FIRST is currently being printed
B:\SECOND is in queue
B:\THIRD is in queue
B:\FOURTH is in queue

A>
```

the command

```
A> PRINT B:SECOND /C
```

will remove the file, as shown here:

```
A> PRINT B:SECOND/C

B:\FIRST is currently being printed
B:\THIRD is in queue
B:\FOURTH is in queue

A>
```

To notify you of the file's deletion, PRINT will display the message

```
File SECOND canceled by operator
```

on your hardcopy listing.

Tip: *To remove several files from the print queue at one time, type all of the file names to remove immediately following the /C qualifier.*

Again, assuming that your print queue contains the files

```
A> PRINT

B:\FIRST is currently being printed
B:\SECOND is in queue
B:\THIRD is in queue
B:\FOURTH is in queue
B:\FIFTH is in queue

A>
```

you can delete the last three queued files by issuing the command

```
A> PRINT B:THIRD /C B:FOURTH B:FIFTH
```

Displaying the files remaining in the queue reveals

```
A> PRINT B:THIRD /C B:FOURTH B:FIFTH

B:\FIRST is currently being printed
B:\SECOND is in queue

A>
```

Tip: *You can use the DOS wildcard characters in conjunction with the PRINT command /C qualifier.*

Assume, for example, that you want to remove all the files in the print queue that contain the extension DAT. You can do this easily with one command, as shown here:

```
A> PRINT /C *.DAT
```

Trap: *If the print queue contains multiple entries of the file to remove, the /C qualifier instructs PRINT to remove all of them.*

Given the queue

```
B:\NUMBERS.DAT is currently being printed
B:\LETTERS.DAT is in queue
B:\PRINT.NTS is in queue
B:\LETTERS.DAT is in queue
B:\PRINT.NTS is in queue
```

the command

```
A> PRINT B:LETTERS.DAT /C
```

removes each occurrence of the file, as shown here:

```
B:\NUMBERS.DAT is currently being printed
B:\PRINT.NTS is in queue
B:\PRINT.NTS is in queue
```

If you need to terminate the file that is currently printing, or if you want to remove all of the files from the queue, use the PRINT /T qualifier.

Tip: *The /T qualifier instructs PRINT to cancel all of the files in the print queue.*

Assuming that your queue contains

```
B:\NUMBERS.DAT is currently being printed
B:\LETTERS.DAT is in queue
B:\PRINT.NTS is in queue
```

the command

```
A> PRINT /T
```

directs PRINT to remove all of the files from the queue. PRINT notifies you of the cancelation by displaying the message

```
All files canceled by operator
```

on your hardcopy listing.

Some printers have their own built-in memory, which buffers the information to be printed. In some cases, although you issue a PRINT /C or PRINT /T command, it may take a moment for the printer to acknowledge the command as it prints the contents of its internal memory.

Since PRINT allows you to add several files to the print queue with one command and to delete several files with one command, it only makes sense that PRINT should allow you add files to and remove files from the queue in one operation.

Tip: *The /P qualifier instructs PRINT to add files to the print queue in a command line that is removing files from the queue with /C.*

Consider the command line

```
A> PRINT OLDFILE /C NEWFILE /P
```

Given this command, PRINT will remove the OLDFILE file from the queue while adding the NEWFILE file. In a similar manner, the command

```
A> PRINT A /C B C /P D
```

will remove files A and B from the queue while adding files C and D.

Remember that the /C qualifier removes the file whose name precedes the qualifier, as well as those that follow. In this case, however, PRINT encounters the /P qualifier, which instructs it to add the file that precedes the qualifier (in this case C) to the print queue, as well as those that follow. PRINT allows you to use /P and /C multiple times throughout your command line.

PRINT Configuration Qualifiers

The first time that you issue the PRINT command, PRINT displays the following message:

```
Resident part of PRINT installed
```

PRINT installs permanently in memory the software that is responsible for managing your printer and the files to print. At the time PRINT installs this software, it determines such items as the target device (LPT1 or COM1), the queue size, and several attributes used for sharing the computer with other DOS commands. As you will see, unless PRINT is told otherwise (by command-line qualifiers), it will use a set of default values.

This section examines the PRINT attributes that you can define with the qualifiers /D, /Q, /B, /M, /S, and /U. Since these qualifiers affect how PRINT configures itself in memory, they are valid only the first time that you invoke the PRINT command.

Using /D to Define The Output Device

When you invoke PRINT for the first time, it displays the prompt

```
Name of list device [PRN]:
```

Remember that you can press ENTER to select the default PRN device as PRINT's target for output or type an alternative device name, such as COM1.

Tip: *The /D qualifier allows you to specify your output device the first time you invoke the DOS PRINT command.*

If you are using a parallel printer attached to LPT1, for example, your command line becomes

```
A> PRINT /D:LPT1
```

If you are using a serial printer connected to COM1, the command line is

```
A> PRINT /D:COM1
```

In either case, since you have specified the desired output device within your command line, PRINT will not display the prompt

```
Name of list device [PRN]:
```


Trap: *If you specify an invalid device name following the /D qualifier in the PRINT command line, PRINT will not install.*

Just as PRINT displays the error message

```
List output is not assigned to a device
```

when you specify an invalid device name at the prompt

```
Name of list device [PRN]:
```

PRINT will display the same error message if you specify an invalid device name in your command line, as in

```
A> PRINT /D:LLT1
```

Using /Q to Increase The Size of the Print Queue

Trap: *By default, PRINT creates a queue large enough to contain ten files.*

If you attempt to place an eleventh file in the queue, PRINT will display the error message

```
PRINT queue is full
```

If you need to create a queue for more than ten files, use the /Q qualifier the first time you invoke PRINT.

Tip: *The /Q qualifier allows you to specify a queue size of up to 32 entries the first time that you invoke the DOS PRINT command.*

The following command, for example, creates a print queue that is large enough to store 32 files:

```
A> PRINT /Q:32
```

As before, this qualifier is valid only the first time that you invoke the PRINT command.

Trap: *Each queue entry consumes 64 bytes of memory.*

The only adverse effect of creating a queue large enough for 32 entries is that it consumes 2048 bytes of memory. However, since most systems today have at least 512K, the memory consumed by 32 queue entries is not significant.

Using /B to Increase The Size of the Print Buffer

When the DOS PRINT command prints a file, it does so by copying a portion of the file from disk into memory before printing it (see Figure 8-1).

When PRINT has printed the contents of the intermediate memory buffer, it reads the next portion of the file into the memory buffer, repeating the process. As stated ealier in this book, disk I/O operations are time-consuming due to the mechanical nature of disks. Therefore, if you can reduce the number of times that DOS must read from a disk, you reduce the amount of time consumed by disk accesses.

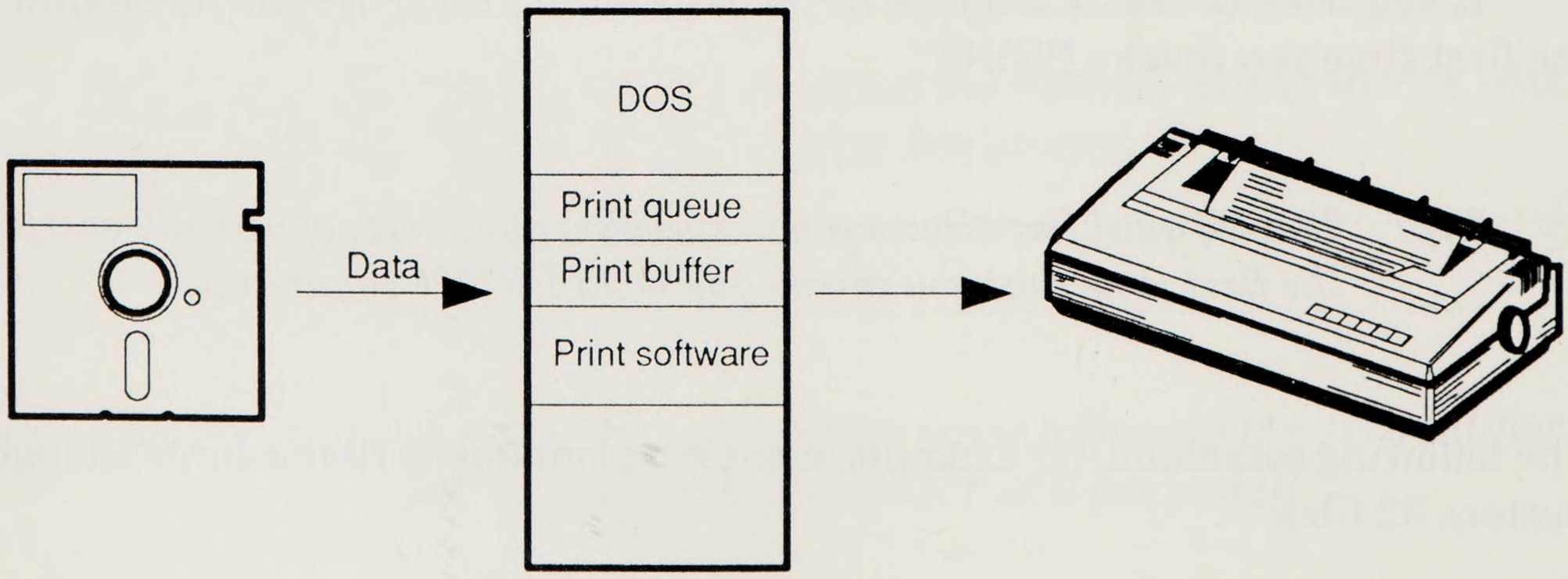


Figure 8-1. Use of the buffer by the PRINT command

Tip: *The /B qualifier allows you to increase the size of the print buffer in memory, which will reduce the number of disk input operations that PRINT must perform to print your file.*

If you increase the size of the intermediate buffer that PRINT uses in memory, you also increase the amount of the file that PRINT can read at one time. In turn, you reduce the number of disk input operations required for printing your file, which improves your overall system performance.

Tip: *PRINT allows you to specify a buffer size from 512 to 16,386 bytes.*

In most cases, you will find that a print buffer of 4096 bytes provides the best overall system performance without consuming too much memory in the process. To create a 4096-byte buffer, issue the command

```
A> PRINT /B:4096
```

If you frequently print very large documents, however, you might consider a larger print buffer size. Increase the size of your print buffer in 512-byte multiples (512, 1024, 2048, and so on).

Increasing PRINT's Processing Time with /M, /S, and /U

Remember that PRINT shares the computer with other DOS activities, allowing your files to print while you continue to execute other commands. This section examines three qualifiers—/M, /S, and /U— each of which influences the amount of time PRINT has to print your files while sharing the computer with DOS.

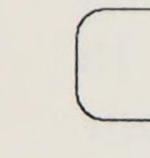
Tip: *The /M qualifier allows you to specify the amount of time that PRINT can retain the computer each time it gets a turn.*

The /M qualifier allows you to specify the number of clock ticks (which occur every 0.0549 seconds) for which PRINT can control the computer with each turn. You can specify a value between 1 and 255. By default, PRINT gets two clock ticks. Most users will get optimal response from a value of 64, for which you issue the command

```
A> PRINT /M:64
```

Trap: *Specifying too large a value for the PRINT /M qualifier will slow your overall system performance, making your computer appear sluggish.*

If you specify a value of /M:255, for example, PRINT will get almost all of the computer's processing time and your keyboard response time will suffer. As stated, most users will find a value of 64 satisfactory.



Tip: *The PRINT /S qualifier specifies the number of times per second that PRINT can control the computer.*

Your computer switches control between your other DOS commands and PRINT at a very rapid rate. The /S qualifier specifies the number of time a second that PRINT can gain control of the computer. Increasing the value associated with /S decreases the number of times PRINT gains control of the computer. Most users keep the default value of 8, but PRINT allows you to specify a value between 1 and 255. In most cases, 8 is sufficient.

If you are using a slow printer, there may be many times when PRINT wastes its turn with the computer because the printer is not ready for more characters.

Tip: *The /U qualifier allows you to specify the number of clock ticks that PRINT can wait if the printer is currently busy.*

Since allowing PRINT to wait for a busy printer ties up your entire system, the default number of clock ticks that PRINT can wait is 1. Using the /U qualifier, you can specify a value between 1 and 255. Most printers are fast enough so that this qualifier is not required. However, you may want to experiment with it.

Once you identify the PRINT command that gives you optimal performance, consider placing the command in the AUTOEXEC.BAT file, which will be discussed later in this book.

Tips and Traps Summary

Getting Started with PRINT

Tip: By default, PRINT sends all of its output to the PRN devie. If you have a parallel printer attached to LPT1, press ENTER at the prompt

```
Name of list device [PRN]:
```

If you have a serial printer, type the device name **COM1** and press ENTER.

Tip: If you are using a serial printer, type **COM1** at the prompt

```
Name of list device [PRN]: COM1
```

Trap: If you specify an invalid device name at the prompt

```
Name of list device [PRN]:
```

PRINT will not install in memory.

Tip: You can print multiple files with one PRINT command line.

Tip: PRINT keeps a list of the files it needs to print. This list is called a queue.

Tip: To display the list of files in the print queue, issue the PRINT command without specifying any files.

Tip: PRINT always prints files in the order in which they were placed in the queue.

Tip: PRINT supports the DOS wildcard characters.

Tip: Not all files are printable.

Trap: If you delete a file currently contained in the print queue, the PRINT command will print unrecognizable characters while attempting to print a nonexistent file.

Tips and Traps Summary (*continued*)

Removing Files from the Print Queue

Tip: The /C qualifier instructs PRINT to remove the specified file(s) from the print queue.

Tip: To remove several files from the print queue at one time, type all of the file names to remove immediately following the /C qualifier.

Tip: You can use the DOS wildcard characters in conjunction with the PRINT command /C qualifier.

Trap: If the print queue contains multiple entries of the file to remove, the /C qualifier instructs PRINT to remove all of them.

Tip: The /T qualifier instructs PRINT to cancel all of the files in the print queue.

Tip: The /P qualifier instructs PRINT to add files to the print queue in a command line that is removing files from the queue with /C.

PRINT Configuration Qualifiers

Tip: The /D qualifier allows you to specify your output device the first time you invoke the DOS PRINT command.

Trap: If you specify an invalid device name following the /D qualifier, in the PRINT command line following the /D qualifier, PRINT will not install.

Trap: By default, PRINT creates a queue large enough to contain ten files.

Tip: The /Q qualifier allows you to specify a queue size of up to 32 entries the first time that you invoke the DOS PRINT command.

Tips and Traps Summary (*continued*)

Trap: Each queue entry consumes 64 bytes of memory.

Tip: The /B qualifier allows you to increase the size of the print buffer in memory, which will reduce the number of disk input operations that PRINT must perform to print your file.

Tip: PRINT allows you to specify a buffer size from 512 to 16,386 bytes.

Tip: The /M qualifier allows you to specify the amount of time that PRINT can retain the computer each time it gets a turn.

Trap: Specifying too large a value for the PRINT /M qualifier will slow your overall system performance, making your computer appear sluggish.

Tip: The PRINT /S qualifier specifies the number of times per second that PRINT can control the computer.

Tip: The /U qualifier allows you to specify the number of clock ticks that PRINT can wait if the printer is currently busy.

9

Customizing DOS

Chapter 6 examined ways to save time and keystrokes by taking advantage of the features that DOS builds into your keyboard. In Chapter 8 you learned how to fine-tune the DOS PRINT command in order to get the best overall system performance. Users are always interested in ways to make their systems perform faster, which in turn makes them more productive. This chapter will examine several DOS characteristics that you can adjust specifically to configure DOS to your personal needs.

Each time DOS starts, it searches the boot disk for a file named CONFIG.SYS. DOS does not provide this file; you must create it.

```
A> DIR  CONFIG.SYS

Volume in drive A is MS330PP01
Directory of  A:\

File not found
```

CONFIG.SYS is a text file containing several one-line entries that allow you to define various system characteristics. If DOS finds the CONFIG.SYS file on your disk, it reads the entries in the file to configure itself in memory. If the file does not reside on disk, DOS uses its default values.

This chapter examines each of the system configuration entries that you can place into the CONFIG.SYS file, discussing the ramifications of each entry to your system as well as providing several recommended values.

Since DOS uses the values that it finds in CONFIG.SYS to configure itself in memory during system startup, the only time DOS uses CONFIG.SYS is when it boots. Therefore, if you create CONFIG.SYS or modify its contents, you must reboot DOS in order for the change to take effect.

Trap: *DOS uses CONFIG.SYS only during system startup. Therefore, you must reboot DOS in order for your changes to CONFIG.SYS to take effect.*

For advanced users, the following list summarizes the DOS CONFIG.SYS entries through DOS version 3.3.

BREAK=	Enables/disables extended CTRL-BREAK processing at system startup.
BUFFERS=	Increases the number of disk buffers that DOS keeps present in memory, which may reduce the number of slow disk drive input and output operations that DOS must perform.
COUNTRY=	Specifies international date, time, and currency formats.
DEVICE=	Installs software to help DOS communicate with a device. This software is called a *device driver*.
DRIVPARM=	Defines the attributes for a block device (disk- or tape-drive device).
FCBS=	Specifies the number of file control blocks (FCB) that DOS will support for older applications (prior to version 2.0).

FILES=	Specifies the number of files that DOS can open at one time.
LASTDRIVE=	Specifies the disk drive letter of the last disk drive that DOS can access.
SHELL=	Specifies the desired command-line interpreter (shell) DOS is to use.
STACKS=	Defines the number of hardware interrupt stacks that DOS will support.

Creating CONFIG.SYS

If you have a word processor, or if you are familiar with EDLIN, you can use either to modify the contents of the CONFIG.SYS file. If not, since CONFIG.SYS is a text file, you can copy its contents from the keyboard, as shown here:

```
A> COPY CON CONFIG.SYS
```

If you are modifying the contents of an existing CONFIG.SYS file, make a copy of your existing file before changing CONFIG.SYS.

Tip: *Never modify CONFIG.SYS without first making a backup copy.*

Before you modify CONFIG.SYS, save the current contents of the file, as shown here:

```
A> COPY CONFIG.SYS CONFIG.SAV
```

Since DOS uses CONFIG.SYS to configure itself in memory, an errant CONFIG.SYS entry can keep DOS from starting. Be sure that you always have a bootable copy of CONFIG.SYS that you can fall back on.

Remember that DOS examines the CONFIG.SYS file only during system startup. If you modify entries in CONFIG.SYS, the changes will not take effect until you reboot.

Enabling Extended CTRL-BREAK Checking

In Chapter 6 you learned that DOS allows you to terminate many applications by pressing the CTRL-BREAK (or CTRL- C) key combination.

```
A> DISKCOPY A: B:

Insert SOURCE diskette in drive A:

Press any key when ready . . .^C

A>
```

By default, each time that DOS reads from your keyboard or writes to your screen display or printer, it checks to see if you have pressed the CTRL-BREAK key combination. However, you may use an application that performs many calculations and disk I/O operations but rarely writes to your screen display or needs keyboard input. In such cases, although you press CTRL-BREAK, it may take DOS some considerable time to acknowledge it. The DOS BREAK command allows you to enable extended CTRL-BREAK checking.

Tip: *To increase the number of times that DOS checks for a CTRL-BREAK, use the BREAK ON command.*

Remember that by default, DOS checks for a CTRL-BREAK only when it reads from the keyboard or writes to your screen display. If you issue the command

```
A> BREAK ON
```

DOS will increase the number of times that it checks for a CTRL-BREAK. DOS will be quicker to recognize that you have entered a CTRL-BREAK, since it is testing for the CTRL-BREAK more often.

Trap: *Enabling extended CTRL-BREAK checking increases your system overhead, which makes your application programs run slower.*

If you issue the command

```
A> BREAK ON
```

DOS will perform additional CTRL-BREAK checking, as just discussed. During this testing, DOS cannot execute your programs, which means that they are running less often, causing them to take longer to execute. Therefore, most users leave extended checking disabled (the default setting). This command exists primarily to assist programmers during the development and testing of programs.

Tip: *To disable extended CTRL-BREAK checking, issue the BREAK OFF command.*

Just as the DOS command

```
A> BREAK ON
```

enables extended CTRL-BREAK checking, the command

```
A> BREAK OFF
```

disables extended CTRL-BREAK checking.

Tip: *To determine the state of extended CTRL-BREAK processing, issue the BREAK command without specifying ON or OFF.*

If you enter the command

```
A> BREAK
```

DOS will display its current state of extended CTRL-BREAK processing—ON or OFF.

```
A> BREAK
BREAK is off

A>
```

As stated, by default, DOS disables extended CTRL-BREAK processing at system startup. If, instead, you want DOS to enable this processing each time it starts, use the CONFIG.SYS BREAK= entry.

Tip: *The CONFIG.SYS BREAK=ON entry enables extended CTRL-BREAK processing at system startup.*

Due to the overhead associated with extended CTRL-BREAK checking, most users keep the default setting of

```
BREAK=OFF
```

Programmers, however, may enable this processing during application testing.

Trap: *If DOS displays the message*

```
Unrecognized command in CONFIG.SYS
```

after you place a BREAK= entry in CONFIG.SYS, you have not specified ON or OFF following the letters "BREAK=".

Reducing the Number of Disk I/O Operations

Due to their mechanical nature, disks are slower than the electronic components that make up your computer. If you reduce the number of disk input/output (I/O) operations that DOS must perform, you can improve the overall performance of your system.

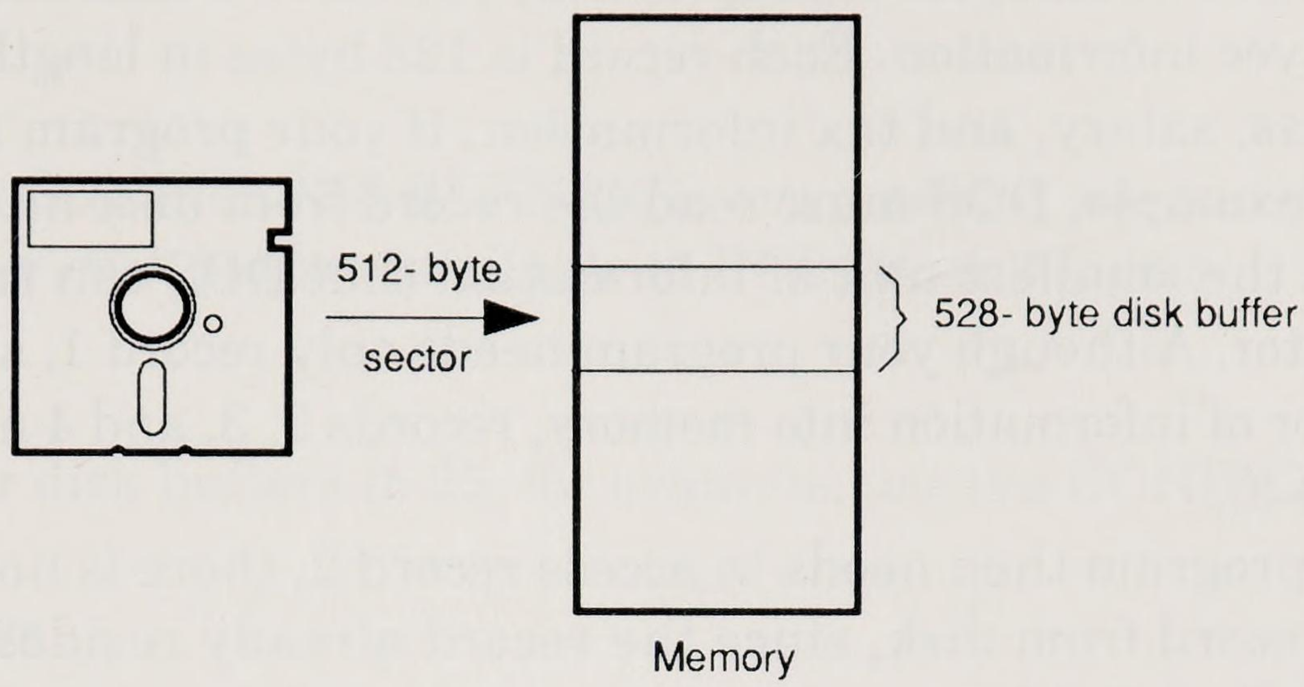


Figure 9-1. Data read from disk is stored in a buffer in memory

Each time DOS reads information from disk, it places the data into an intermediate *buffer* in memory, as shown in Figure 9-1. Each of these intermediate buffers is 528 bytes in length. As stated in Chapter 4, DOS stores the information that you place on disk in 512-byte sectors (see Figure 9-2).

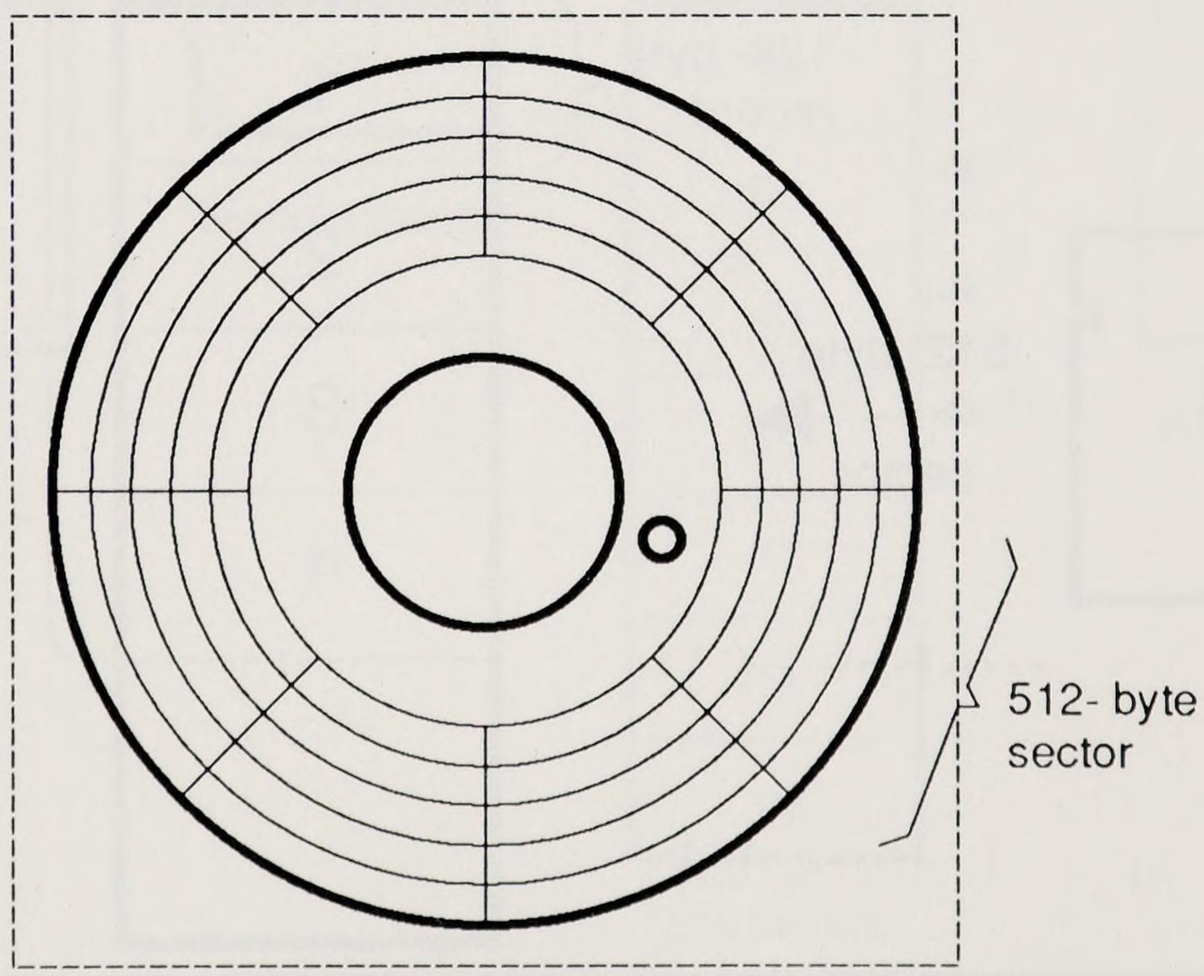


Figure 9-2. DOS stores data on disk in 512-byte sectors

A *sector* is the smallest unit of information that DOS can transfer to or from your disk. Assume, for example, that you have a database program that tracks employee information. Each record is 128 bytes in length and contains a name, address, salary, and tax information. If your program needs to update record 1, for example, DOS must read the record from disk into memory. Keep in mind that the smallest unit of information that DOS can transfer is a 512-byte disk sector. Although your program needs only record 1, since DOS transferred a sector of information into memory, records 2, 3, and 4 are also available (see Figure 9-3).

If your program then needs to access record 2, there is no reason for DOS to read the record from disk, since the record already resides in memory. By using disk buffers, DOS has reduced the number of slow disk I/O operations.

Tip: *By increasing the number of disk buffers available in memory, you increase the likelihood of DOS finding the desired information in memory, which reduces the number of slow disk input operations DOS must perform.*

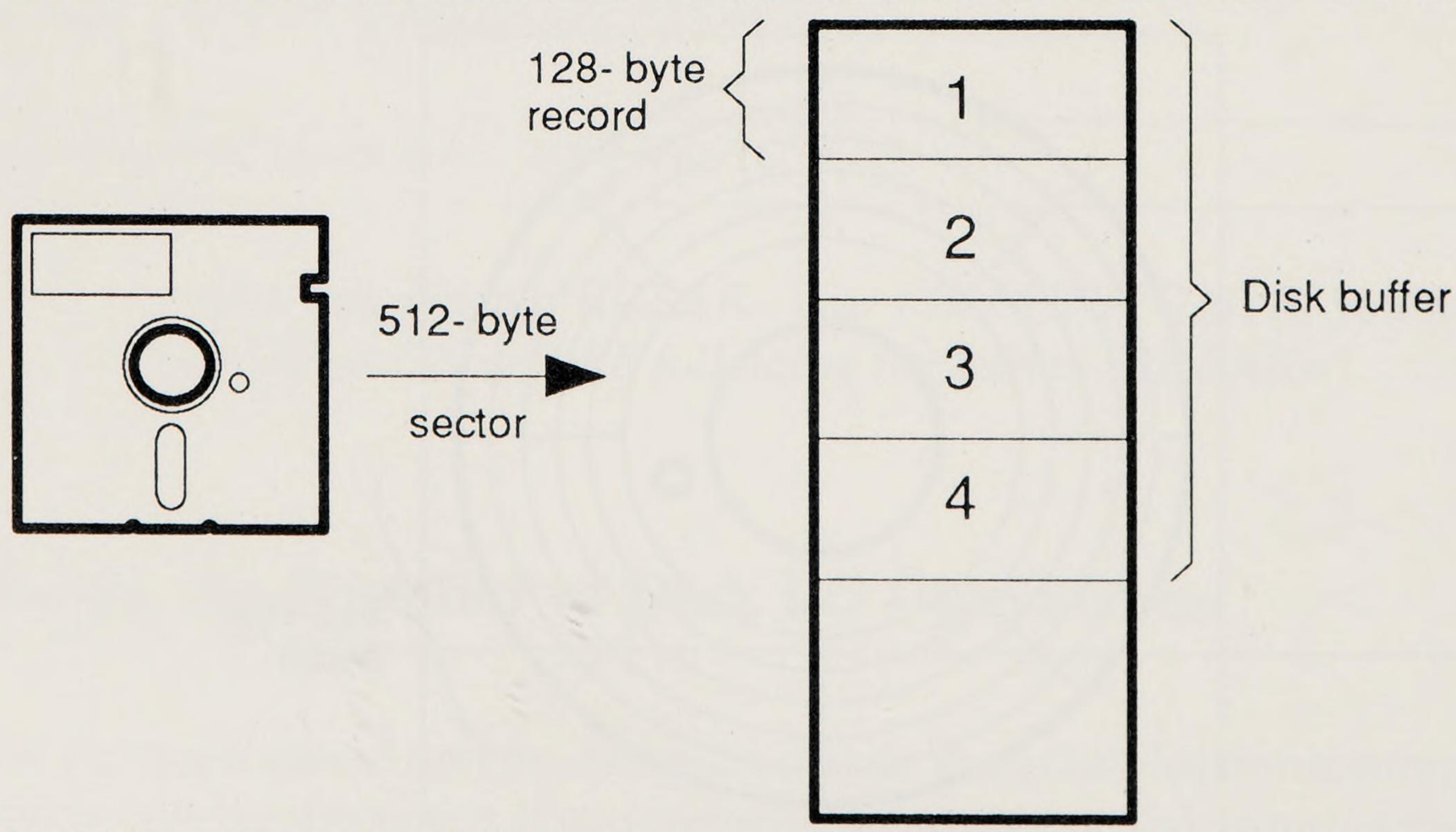


Figure 9-3. Database records read from disk into memory

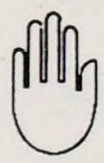
To increase the number of disk buffers available to DOS, use the BUFFERS= entry in CONFIG.SYS.

Tip: *The CONFIG.SYS BUFFERS= entry allows you to define the number of disk buffers available to DOS (from 2 to 255).*

To increase your disk buffers to 25, for example, use the CONFIG.SYS entry

```
BUFFERS=25
```

Before DOS reads information from your disk, it searches the 25 disk buffers in memory for the desired information (see Figure 9-4).



Trap: *If you make the number of disk buffers available to DOS too large, you will decrease your overall system performance.*

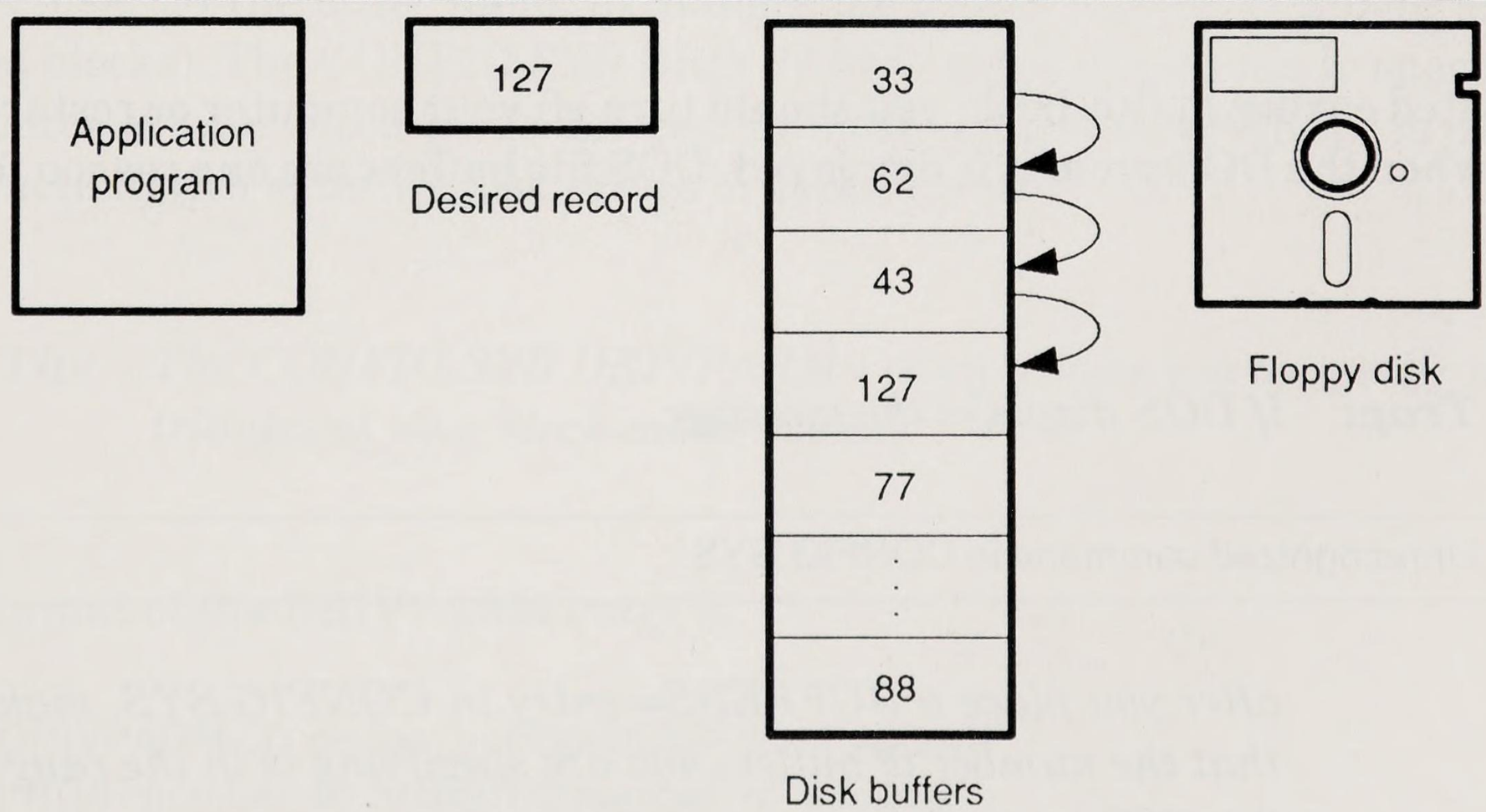


Figure 9-4. Record found in disk buffer

Remember that before DOS reads information from disk, it searches the disk buffers in memory for the data. If DOS must continually traverse these buffers unsuccessfully, it is wasting time. Therefore, too many disk buffers can be as detrimental as too few buffers. Since you run many different types of programs in your system, it is difficult to fine-tune this parameter for a particular program. Instead, you should set it to a constant size that is suitable for all applications.

Tip: *For most applications, 25 disk buffers is sufficient.*

As discussed earlier, the smallest unit of information that DOS can transfer to or from disk is a 512-byte sector. When your program writes information to disk, the information is not actually recorded on the disk storage media until DOS has 512 bytes of information (or the file is closed). Until that point, DOS keeps this information in a disk buffer.

Trap: *If you turn off your computer or restart DOS while there is data in a disk buffer, the information in the buffer is not recorded to disk and is lost.*

As stated earlier in this book, you should turn off your computer or restart DOS only when the DOS prompt is displayed. DOS file buffers are one reason for this rule.

Trap: *If DOS displays the message*

Unrecognized command in CONFIG.SYS

after you place a BUFFERS= entry in CONFIG.SYS, make sure that the number of buffers you are specifying is in the range from 2 to 255.

Defining Country-Specific Information

With the success abroad of the personal computer and DOS, additional support has been provided to international users. DOS provides various international keyboard templates; character sets; and date, time, and currency formats.

Tip: *The CONFIG.SYS COUNTRY= entry allows you to select international date, time, and currency formats.*

The chapter on international DOS support, Chapter 19, will explain how COUNTRY= entry works.

Setting Block Device Attributes

Devices are categorized as either *block mode* or *character mode,* depending on the manner in which DOS communicates with them. A parallel printer, for example, receives information from DOS a character at a time, as shown in Figure 9-5. A disk, on the other hand, exchanges data with DOS in 512-byte sectors (called blocks). The CONFIG.SYS DRIVPARM= entry allows you to specify the attributes of your block-mode devices. In most cases, DRIVPARM= is used in conjunction with external disk drives or tape drives.

Tip: *The CONFIG.SYS DRIVPARM= entry allows you to specify the attributes of your block-mode devices.*

The format of the DRIVPARM entry is

```
DRIVPARM=/D:device_number [/C] [/F:form_factor]
[/N] [/H:number_of_heads] [/S:number_of_sectors]
[\T:number_of_tracks]
```

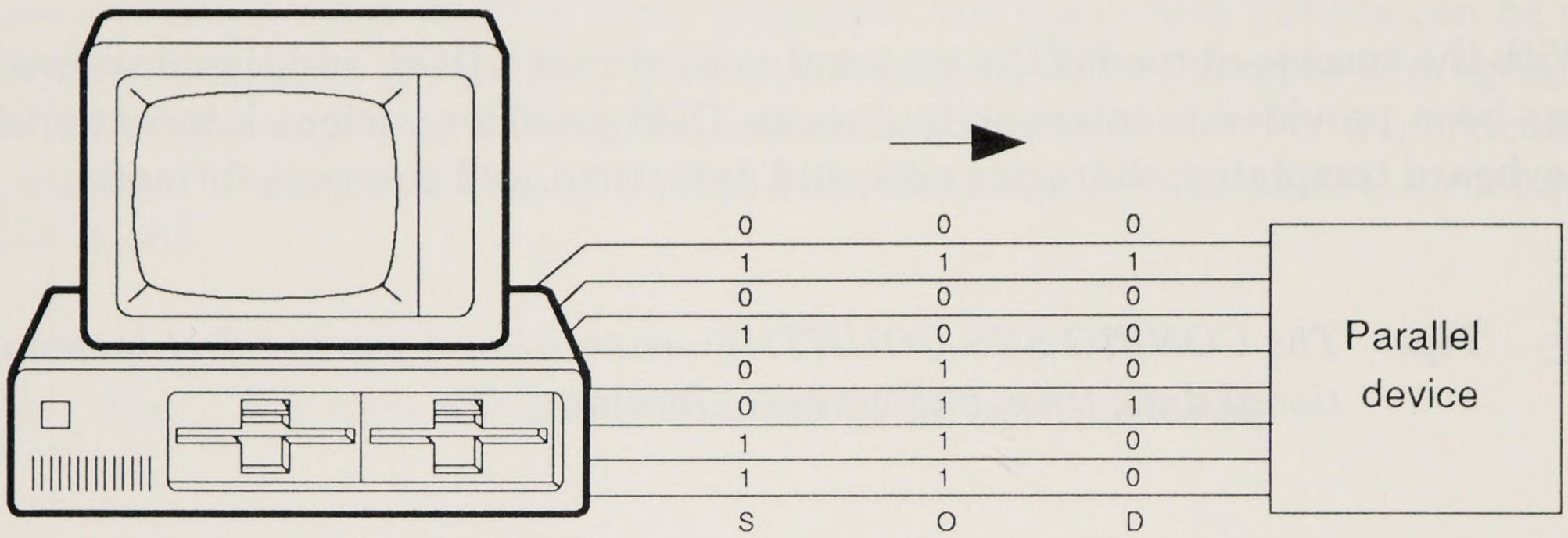


Figure 9-5. Character mode transmission

Items that appear within brackets are optional. The following list describes these *parameters* to DRIVPARM.

/D:device_number is the physical drive number (0 to 255) of the drive. A is 0, B is 1, C is 2, and so on.

/C specifies that doorlock support is required. This means that the device driver will be able to tell whether a disk drive latch is closed or open.

/F:form_factor specifies the type of device:

0	160/180K
0	320/360K
1	1.2MB
2	720K (3 1/2-inch)
3	8-inch single-density
4	8-inch double-density
5	Hard disk
6	Tape drive
7	1.44MB (3 1/2-inch)

/*N* states that the device is nonremovable.

/*H:number_of_heads* specifies the maximum number of recording heads, from 0 to 99. The default is 2.

/*S:number_of_sectors* specifies the number of sectors per track, from 1 to 99. The default is 9.

/*T:number_of_tracks* specifies the number tracks per side, from 1 to 999.

This entry, for example, sets the parameters for a tape drive (form factor 6) to nine tracks with ten sectors per track:

```
DRIVPARM=/D:4 /F:6 /H:1 /S:10 /T:9
```

Most users will never require the DRIVPARM= entry. If you do, refer to the documentation that accompanied your block-mode device, which should contain the correct DRIVPARM= entry.

Trap: *If DOS displays the message*

```
Unrecognized command in CONFIG.SYS
```

after you include a DRIVPARM= entry in CONFIG.SYS, you have very likely included an invalid value in the list of arguments.

Supporting DOS Applications Prior To Version 2.0

Prior to DOS version 2.0, application programs used a structure called a *file control block* (*FCB*) for disk read and write operations. (Newer DOS applications use file handles.) DOS provides minimal support for file control blocks. If you

are running old DOS applications, you may encounter errors during disk read and write operations. If this occurs, try using the CONFIG.SYS FCBS= entry.

Tip: *The CONFIG.SYS FCBS= entry supports older DOS applications that use file control blocks.*

The format of the FCBS= entry is

```
FCBS=open_files,cant_close
```

where the following is true:

open_files specifies the maximum number of files that DOS can open at one time using file control blocks. This value must be in the range from 1 to 255. The default is 4.

cant_close specifies the number of files that DOS cannot automatically close in order to access other files. This value must be in the range from 0 to *open_files*. The default is 0.

Most users will never require the FCBS= entry. If your computer encounters errors while running older software programs, try the entry

```
FCBS=20,4
```

Trap: *If DOS displays the message*

```
Unrecognized command in CONFIG.SYS
```

after you place an FCBS= entry in CONFIG.SYS, you have probably specified an invalid value for the number of FCBs that DOS will support. Specify a value in the range 1 to 255.

Determining the Number Of Files That DOS Can Open

As just discussed, newer versions of DOS (since 2.0) use a structure called a *file handle* for disk input and output operations. By default, DOS provides eight files for use in your programs.

Trap: *Although DOS provides eight file handles by default, it predefines five of them, restricting your applications to opening only three files at one time.*

Chapter 7 examined DOS I/O redirection. Remember that, by default, DOS writes a command's output to the screen display, while obtaining the command's input from the keyboard. DOS names your screen *stdout* (for standard output device) and names your keyboard *stdin* (for standard input device), as shown in Figure 9-6.

DOS uses two of the available file handles for your screen and keyboard. In addition, it assigns a file handle to your printer (stdprn); your serial port (stdaux); and the destination for error messages (stderr), which usually points

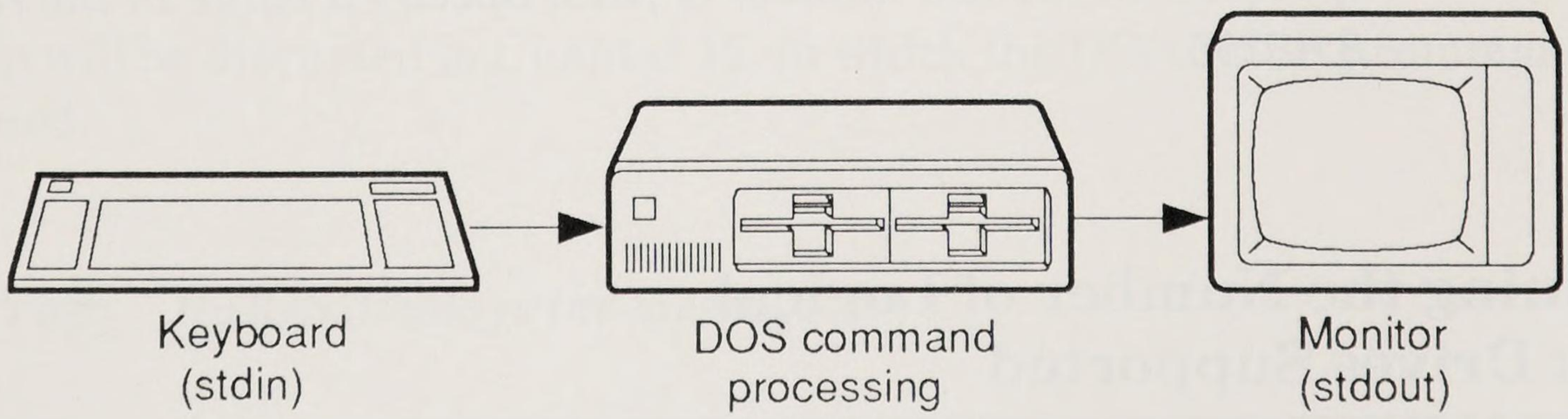


Figure 9-6. DOS names your screen stdout and your keyboard stdin

to your screen. (By sending error messages to stderr, DOS prevents them from being redirected by pipe or I/O redirection operators.)

To the user, the fact that DOS assigns file handles to these devices is not as important as the fact that as a result of assigning handles to these devices, your applications can open only three files at a time. Many programs, specifically database programs, require that many files be open at one time. If these programs are restricted by DOS to three files, the programs will fail to work.

Tip: *To increase the number of files that DOS can open at one time, use the CONFIG.SYS FILES= entry.*

The FILES= entry allows you to specify the maximum number of files that DOS applications can open at one time. This value must be in the range from 8 to 255. In most cases, you will find 20 file handles sufficient:

```
FILES=20
```

Trap: *If DOS displays the error message*

```
Unrecognized command in CONFIG.SYS
```

after you place a FILES= entry in CONFIG.SYS, you have likely specified an invalid number of files. Specify a value in the range 8 to 255.

Defining the Number of Logical Disk Drives Supported

Later in this book several DOS "pretender" commands are examined, which make your system appear to have more disk drives than are physically attached to your computer. The CONFIG.SYS LASTDRIVE= entry allows you to specify the letter of the last disk drive that DOS will support. By default, DOS supports drives A through E.

Tip: *The CONFIG.SYS LASTDRIVE= entry allows you to specify the letter of the last disk drive that DOS will support.*

To support disk drives A through I, for example, your CONFIG.SYS entry becomes

```
LASTDRIVE=I
```

Trap: *DOS will not allow LASTDRIVE to hide existing disk drives.*

If your system has two floppy-disk drives, A and B, and a fixed-disk drive, C, the entry

```
LASTDRIVE=B
```

will not hide drive C from the user.

Tip: *To disable the pretender command SUBST, you can use LASTDRIVE to prevent DOS from creating logical disk drives.*

This tip will be discussed in Chapter 15, in which the DOS SUBST command is examined.

Trap: *If DOS displays the message*

```
Unrecognized command in CONFIG.SYS
```

after you have modified the LASTDRIVE= entry in CONFIG.SYS, make sure that you have specified a disk drive in the range A through Z.

Defining the Command-Line Interpreter

By default, DOS uses COMMAND.COM as its *command-line interpreter.* This file contains the internal DOS commands and is responsible for displaying the DOS prompt and processing your commands. However, some advanced users may choose to install their own command interpreter (or *shell*). The SHELL= CONFIG.SYS entry allows you to do so.

Tip: *The SHELL= entry allows you to specify a command interpreter as an alternative to COMMAND.COM.*

The format of the SHELL= entry is

```
SHELL=[path]filename.ext [options]
```

where the following is true:

[*path*] *filename.ext* tells DOS the name of the file containing the desired command interpreter, as well as where the file resides.

[*options*] are optional qualifiers specific to the command interpreter.

Tip: *Fixed-disk users can remove the COMMAND.COM file from the root directory by using the SHELL= entry in CONFIG.SYS.*

If you are using DOS subdirectories (discussed later in Chapters 13 and 14), you can place the COMMAND.COM file into a subdirectory. Assuming that your DOS files reside in a subdirectory called DOS, place the COMMAND.COM file in that directory, and then place the entry

```
SHELL=C:\DOS\COMMAND.COM /P
```

in CONFIG.SYS. In this command line, the /P qualifier is critical. Without /P, COMMAND.COM will not execute the AUTOEXEC.BAT file (if that file exists) nor display the DATE and TIME prompts.

The next chapter in this book discusses a storage facility called the DOS environment, which allows you to store information for use in conjunction with several advanced DOS commands. If you are using the SHELL= entry in conjunction with COMMAND.COM, DOS allows you to specify a size for the DOS environment from 160 to 32,768 bytes. By default, DOS sets aside 160 bytes for the environment, which is not always adequate. This entry, for example, creates an environment of 4096 bytes:

```
SHELL=C:\DOS\COMMAND.COM /P /E:4096
```

Trap: *If DOS displays the message*

```
Bad or missing command interpreter
```

the file name that you have specified for your command interpreter does not exist on disk.

Creating a Hardware Interrupt Stack

Certain events that occur in your computer are so important that DOS must quit processing its current task to handle the interruption. So that DOS remembers the task that it was performing when it was interrupted and can continue where it left off, DOS places the information that it needs in an area of memory called the *stack*. If these hardware interrupts occur too often, it is possible for DOS to run out of stack space. If this occurs, DOS will display the message

```
FATAL: Internal Stack Failure, System Halted
```

and your system will hang, forcing you to reboot. If this happens, use the CONFIG.SYS STACKS= entry.

Tip: *The CONFIG.SYS STACKS= entry allows you to increase the size of the DOS stack. This reduces the likelihood of DOS running out of stack space if successive hardware interrupts occur.*

The format of the STACKS= entry is

```
STACKS=number_of_stacks,stack_size
```

where the following is true:

number_of_stacks specifies the number of stack frames that DOS will support, from 0 to 64. The default is 0 for the PC, PC-XT, and PC portable. Other systems use a default value of 128.

stack_size specifies the size of each stack frame in bytes, from 0 to 512. By default, DOS uses 0 for the PC, PC-XT, and PC portable. Other systems use a default of 128.

This entry, for example, sets aside ten stack frames of 256 bytes each:

```
STACKS=10,256
```

Unless your system encounters a stack error, displaying

```
FATAL: Internal Stack Failure, System Halted
```

you won't need to use this entry.

Trap: *If DOS displays the message*

```
Unrecognized command in CONFIG.SYS
```

after you place a STACKS= entry in CONFIG.SYS, make sure that your stack frame (0-64) and stack size (0-512) values are correct.

Installing a Device Driver

All of the hardware devices on your system—including the keyboard, screen display, and disk drives—require software to allow DOS to communicate with them. This software is called a *device driver.* Since these devices are standard to all personal computers, DOS builds their device drivers into DOS itself. If you purchase a mouse or other nonstandard hardware device such as a plotter, in most cases you will have to inform DOS about the device's driver. The CONFIG.SYS DEVICE= entry allows you to install a device driver. As you will see in this section, DOS provides several device drivers for your use.

Tip: *The CONFIG.SYS DEVICE= entry instructs DOS to install a device driver in memory.*

The format of the DEVICE= entry in CONFIG.SYS is

```
DEVICE=[path]filename.ext [options]
```

where the following is true:

[*path*] *filename.ext* tells DOS the name of the device driver to install, as well as where the driver resides on disk.

[*options*] are optional qualifiers specific to the device driver.

Tip: *Most device drivers have the extension SYS.*

If you examine your Program Disk and Supplemental Programs Disk (for DOS version 3.3), you will find the following device-driver files:

ANSI.SYS	Provides enhanced keyboard and screen display processing.
DISPLAY.SYS	Contains code-page support for your screen display.

DRIVER.SYS	Contains device-driver support for external floppy-disk drives.
KEYBOARD.SYS	Contains foreign keyboard templates for international DOS support.
PRINTER.SYS	Contains code-page support for your screen display.
RAMDRIVE.SYS	Contains a device driver for a virtual- or RAM-disk drive.

While examining DOS international considerations in Chapter 19, you will learn about the DISPLAY.SYS, KEYBOARD.SYS and PRINTER.SYS files. This discussion will concentrate on DRIVER.SYS, RAMDRIVE.SYS, and ANSI.SYS.

Supporting External Floppy-Disk Drives

All of the systems examined so far in this book have been either single-floppy-disk systems with drive A, or dual-floppy-disk systems with drives A and B, as shown in Figure 9-7. However, some systems have an external floppy-disk drive, as shown in Figure 9-8.

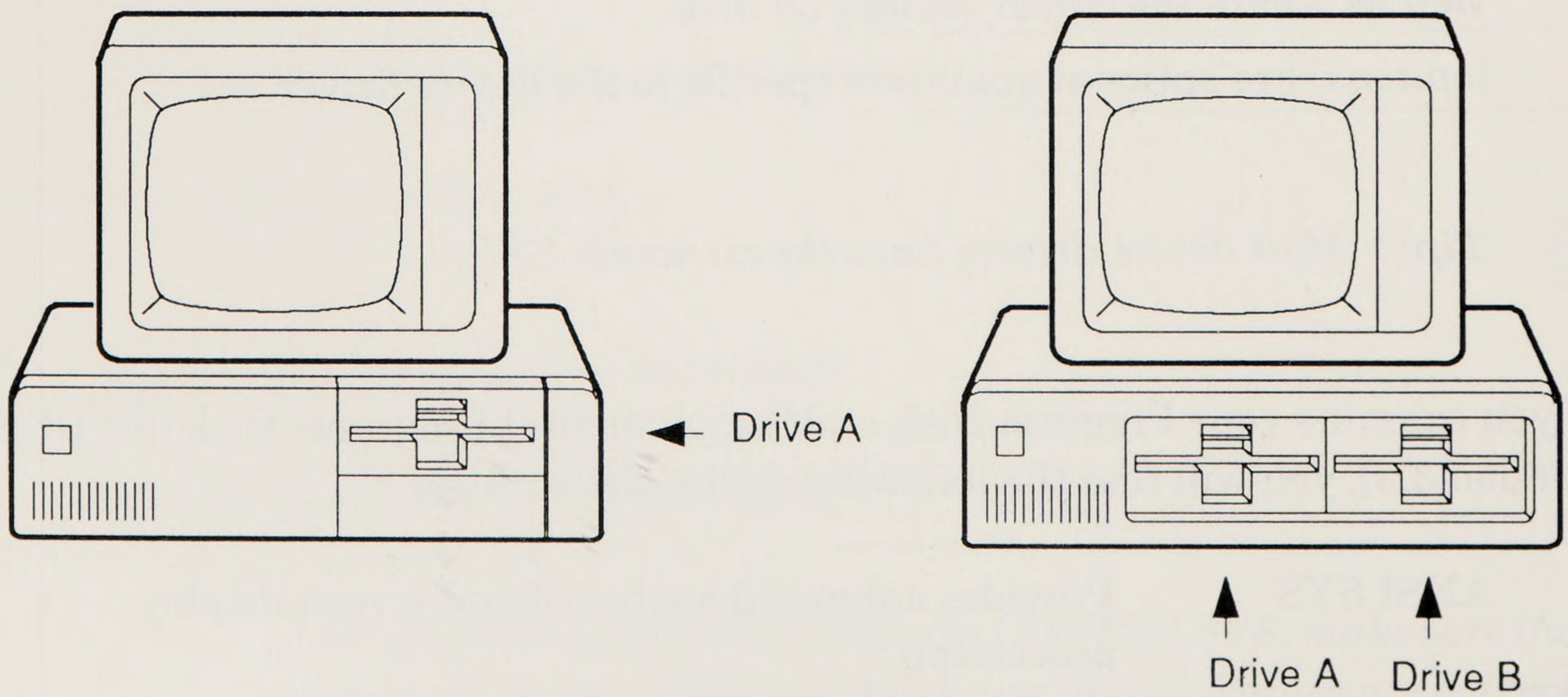


Figure 9-7. Systems with internal floppy-disk drives

Since an external floppy-disk drive is not a "standard" device, you must install a device driver in order for DOS to communicate with it. The DRIVER.SYS file contains a device driver for external floppy-disk drives.

Tip: *DRIVER.SYS contains the DOS device drivers for external floppy-disk drives.*

The format of the DRIVER.SYS entry is

```
DEVICE=[path]DRIVER.SYS /d:drive_number [/C]
[/F:form_factor] [/H:number_of_heads]
[/N] [/S:number_of_sectors] [/T:number_of_tracks]
```

where the following is true:

/d:drive_ number is the physical drive number (0 to 255) of the drive. A is 0, B is 1, C is 2, and so on.

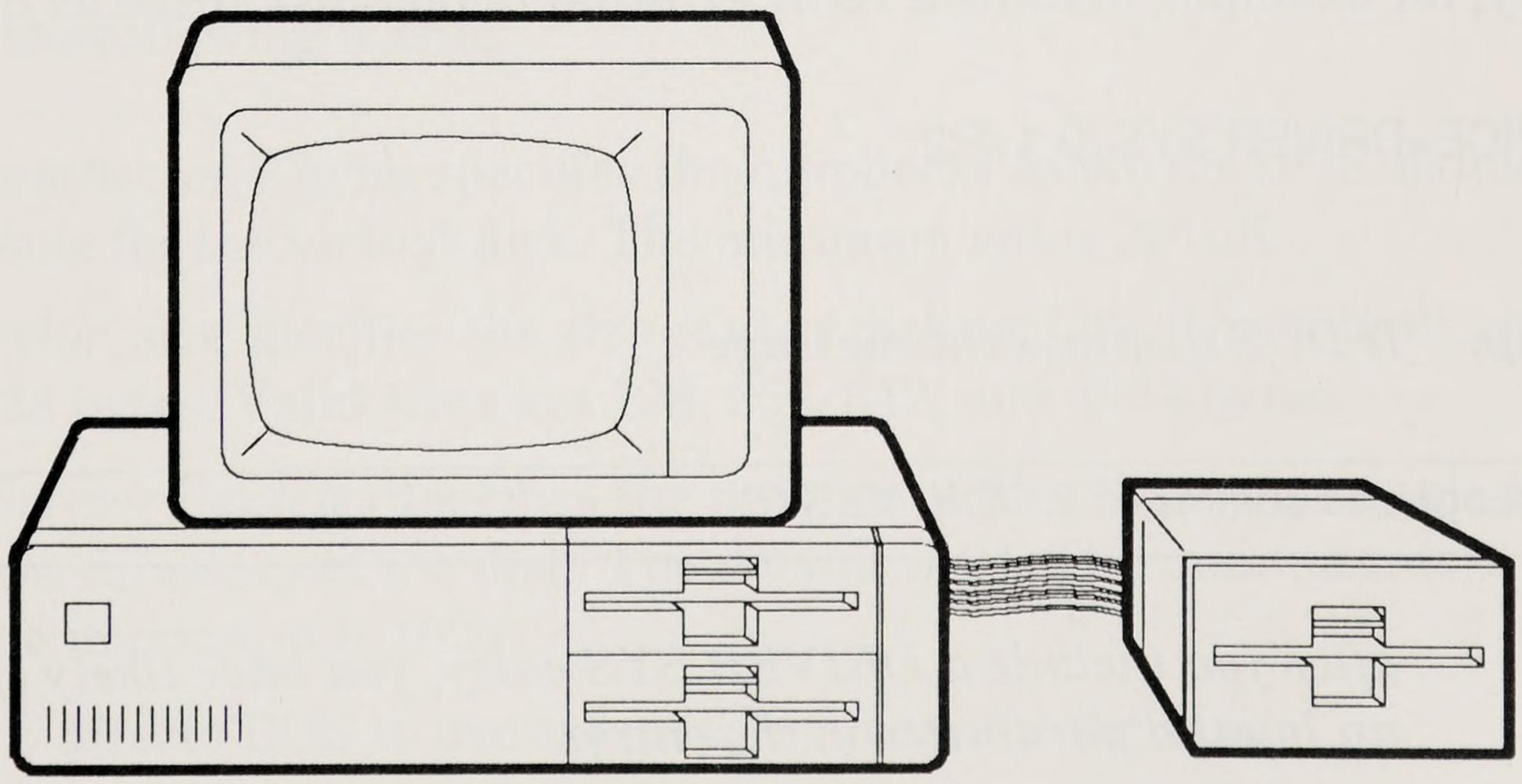


Figure 9-8. A system with an external floppy-disk drive

/C specifies that doorlock support is required. This means that the device driver will be able to tell whether a disk drive latch is closed or open.

/F:form_factor specifies the type of device:

0	160/180K
0	320/360K
1	1.2MB
2	720K (3 1/2-inch)
3	8-inch single-density
4	8-inch double-density
5	Hard disk
6	Tape drive
7	1.44MB (3 1/2-inch)

/H:number_of_heads specifies the maximum number of recording heads, from 0 to 99. The default is 2.

/N states that the device is nonremovable.

/S:number_of_sectors specifies the number of sectors per track, from 1 to 99. The default is 9.

/T:number_of_tracks, specifies the number of tracks per side, from 1 to 999.

This entry, for example, installs a 720K external floppy-disk drive as drive B:

```
DEVICE=DRIVER.SYS /D:1 /F:2
```

Trap: *If DOS displays the message*

```
Unrecognized command in CONFIG.SYS
```

after you include a DRIVER.SYS entry, you have likely specified an invalid parameter in the entry.

Installing a Virtual-Disk Drive in RAM

The DOS RAMDRIVE.SYS device driver allows you to install a temporary disk drive in memory, which functions as a very fast disk drive. Because this is not a physical disk drive attached to your system, it is called a *virtual disk*. Virtual disks have the following characteristics:

- They are very fast. Because virtual disks reside in memory, as opposed to being mechanical devices, they are up to ten times as fast as your fixed disk.
- They lose their contents when you turn off your computer. Again, because virtual disks reside in memory, the information they contain is lost when you turn off your system or restart DOS.

Virtual disks are also called *RAM disks*. Your computer's memory is called *random access memory* or *RAM*. Since virtual disks reside in RAM, they are called *RAM drives*.

The format of RAMDRIVE.SYS is

```
DEVICE=RAMDRIVE.SYS [number_of_Kbytes] [sector_size]
[directory_entries] [/E or /A]
```

where the following is true:

number_of_Kbytes specifies the number of kilobytes DOS should set aside for the virtual disk. The minimum value is 16K.

sector_size specifies the size of your disk sectors. The default value is 128 bytes. Valid sizes are 128, 256, 512, and 1024 bytes.

directory_entries specifies the number of files that you can place in the root directory of the disk. The default is 64. The minimum is 4, while the maximum is 1024.

/E directs DOS to use extended memory for the disk.

/*A* directs DOS to use extended memory that meets the Lotus/Intel/Microsoft expanded-memory specification for the disk.

The following entry, for example, creates a RAM drive using the default configuration of 64K, 128-byte sectors, and 64 directory entries:

```
DEVICE=RAMDRIVE.SYS
```

In this case, when DOS starts, it will display

```
Microsoft RAMDrive version 1.19 virtual disk D:
Disk size: 64K
Sector size: 128 bytes
Allocation unit: 1 sectors
Directory entries: 64
```

while informing you that it has successfully installed your RAM disk.

At this point, you must be sure that the RAMDRIVE.SYS file is on the disk that you will use to start DOS. In many cases, RAMDRIVE.SYS resides on the DOS Supplemental Programs Disk, so you will need to copy it to the disk you use to boot your system.

Trap: *Depending on your version of DOS, the RAMDRIVE.SYS device driver may be named VDISK.SYS.*

If your system uses VDISK.SYS, use this driver as discussed in the section discussing your CONFIG.SYS entry.

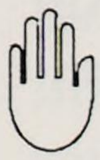
```
DEVICE=VDSISK.SYS
```

Tip: *Once installed, you access your virtual disk with the disk drive letter displayed in the message*

```
Microsoft RAMDrive version 1.19 virtual disk D:
```

In this case, the virtual-disk drive is D. To select this drive, simply use

```
A> D:
D>
```



Trap: *RAM drives lose their contents when DOS restarts.*

Copy the COMMAND.COM file to drive D.

```
A> COPY  COMMAND.COM  D:
            1 File(s) copied
```

A directory listing of drive D now reveals

```
A> DIR  D:

Volume in drive D is MS-RAMDRIVE
Directory of  D:\

COMMAND          COM          25307    5-11-88  2:05p
            1 File(s) 37248 bytes free
```

Next, using the CTRL-ALT-DEL key combination, restart DOS. When DOS restarts, again select drive D. A directory listing reveals that no files remain:

```
A> DIR  D:

Volume in drive D is MS-RAMDRIVE
Directory of  D:\

File not found
```

When you restarted DOS, your RAM drive lost the files it contained (in this case, COMMAND.COM).

Many floppy-disk users copy their most commonly used commands to a RAM drive to avoid continually changing floppy disks.

Tip: *If you place data files in your RAM drive, back up the files to a fixed or floppy disk on a regular basis.*

Placing command files on a RAM drive is normally safe because you have copies of these files on disk. Be careful, however, if you place data files on your RAM drive. If for some reason you have to restart DOS, you will likely lose the contents of these files.

Trap: *If DOS displays the message*

```
Microsoft RAMDrive version 1.19 virtual disk D:
RAMDrive: Invalid parameter
```

when your system restarts, make sure that your RAM-DRIVE.SYS entries are correct.

Having Fun with ANSI.SYS

The ANSI.SYS device driver is software that allows you to control your screen display output and enhance your keyboard capabilities. In the past, ANSI.SYS was most commonly used by programmers, but as you will see, it provides tremendous capabilities for the user, such as changing screen color and function-key assignments.

Tip: *The ANSI.SYS device driver enhances your screen output and keyboard input capabilities.*

To install the ANSI.SYS device driver, enter

```
DEVICE=ANSI.SYS
```

As stated, once you modify a CONFIG.SYS entry, you must reboot DOS. Once DOS restarts, you can issue ANSI escape sequences to achieve the desired results. An *ANSI escape sequence* is a series of characters that begins with the *escape character* (an ASCII 27). For example, the ANSI escape sequence Esc[2J instructs DOS to clear the screen display. If you are using a word processor, refer to its documentation to find out how to enter the escape character. Since you have EDLIN on your DOS Program Disk, as you can see by the command

```
A> DIR  EDLIN.COM

Volume in drive A is MS330PP01
Directory of  A:\

EDLIN          COM          7495     7-24-87  12:00a
            1 File(s)     4096 bytes free
```

use EDLIN to create a file containing an ANSI escape sequence:

```
A> EDLIN  B:CLEAR.SCR
New File
*
```

At the EDLIN prompt (*), type **I** and press ENTER. EDLIN will prompt you to enter line 1, as shown here:

```
A> EDLIN  B:CLEAR.SCR
New File
*I
            1:*
```

To enter the escape character, hold down the CTRL key and type **V**. EDLIN will respond with

```
A> EDLIN  B:CLEAR.SCR
New File
*I
                1:^V
```

The second part of the escape character requires you to enter the left-bracket character, [:

```
A> EDLIN  B:CLEAR.SCR
New File
*I
                1: *^V[
```

You have now entered this much of the escape sequence:

```
^V[      [ 2J
 \_/     \_/
Esc      [ 2J
```

Now type **[2J**, as shown here:

```
A> EDLIN  B:CLEAR.SCR
New File
*I
                1: *^V[[2J
```

The letter "J" must be in uppercase. Press ENTER, and EDLIN will respond with a prompt for line 2:

```
A> EDLIN  B:CLEAR.SCR
New File
*I
                1: *^V[[2J
                2: *
```

Since you don't have a second line, simply press CTRL-C.

```
A> EDLIN  B:CLEAR.SCR
New File
*I
          1: *^V[[2J
          2: *^C
*
```

To exit EDLIN, saving the file that you just created, type **E** and press ENTER.

```
A> EDLIN  B:CLEAR.SCR
New File
*I
          1:^V[[2J
          2:^C
*E

A>
```

At the DOS prompt, issue the command

```
A> TYPE  B:CLEAR.SCR
```

DOS will respond by clearing your screen display.

The ANSI escape sequence Esc[2J directed DOS to erase the contents of your screen, placing the cursor in the upper-left corner. Using ANSI escape sequences is that easy! Let's see what else they can do.

The escape sequence Esc[*n*m allows you to set your screen color and video attributes. For example, Esc[44m sets your screen's background color to blue. Using EDLIN, create a file called BLUEBACK.SCR that changes your screen's background color, as shown here:

```
A> TYPE  B:BLUEBACK.SCR
```

The values shown in Table 9-1 are valid for *n* in the escape sequence Esc[*n*m.

Table 9-1. Values for n in the Escape Sequence Esc[nm

Value	Function
0	Default attributes
1	Bold text
2	Low-intensity text
3	Italic text
5	Blinking text
6	Rapid blinking text
7	Reverse video on
8	Concealed text
30	Black foreground
31	Red foreground
32	Green foregound
33	Yellow foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground
40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background
48	Subscript
49	Superscript

DOS allows you to set multiple screen attributes at one time by separating each attribute by a semicolon. For example, the escape sequence Esc[33;44m sets the background to blue and the foreground to yellow. By placing escape sequences in your text files, you can create very neat effects. For example, the file

```
A> EDLIN  B:MESSAGE.TXT
New File
*I
                1:^V[[5m
                2:WARNING!
                3:^V[[0m
                4:FORMAT overwrites your target disk.
                5:^C
*E

A>
```

will display the message

```
WARNING!
```

blinking on and off when the user types the file's contents.

In addition to setting your screen attributes, the ANSI.SYS device driver allows you to control the cursor position and hence where DOS writes output on your screen display. Table 9-2 summarizes the ANSI cursor-control commands.

The following file clears the screen and then displays a message:

```
A> EDLIN  B:NEW.MSG
New File
*I
                1:^V[[2J
                2:^V[[10;35HWARNING!
                3:^V[[11;35HFORMAT overwrites your target disk.
                4:^C
*E

A>
```

The message is displayed in the center of your screen:

WARNING!
FORMAT overwrites your target disk.

The following complex escape sequence allows you to define a key on your keyboard:

```
Esc[0;68;"Keyboard redefinition"p
```

This escape sequence allows you to assign a key on your keyboard to a commonly used command. The following example assigned the F10 function key to DIR:

```
A> EDLIN  B:KEY.DIR
New File
*I
          1:^V[[0;68;"DIR"p
          2:^C
*E

A> TYPE  B:KEY.DIR

A>
```

Table 9-2. ANSI Cursor-Control Commands

Sequence	Function	Example
Esc[Row;ColumnH	Set cursor at row, column	Esc[10;1H
Esc[*n*A	Move cursor up *n* rows	Esc[4A
Esc[*n*B	Move cursor down *n* rows	Esc[4B
Esc[*n*C	Move cursor right *n* columns	Esc[10C
Esc[*n*D	Move cursor left *n* columns	Esc[10D
Esc[6n	Get device status report	
Esc[s	Save cursor position	
Esc[u	Restore cursor position	
Esc[2J	Clear screen display	

When you subsequently press F10, DOS will display DIR on your screen, allowing you to simply press ENTER.

Trap: *If your screen displays strange numbers or characters when you type a file containing an ANSI escape sequence, either your sequence is incorrect or you do not have the ANSI.SYS driver correctly installed.*

If this occurs, make sure CONFIG.SYS contains the entry

```
DEVICE=ANSI.SYS
```

and you have rebooted DOS for the entry to take effect. Next, make sure that you have the correct number of left brackets in your escape sequence.

The ANSI.SYS device driver allows you to control your screen output and keyboard input. Many printers also support various escape sequences. Using EDLIN in this manner, you can create simple text files that you later print to direct your printer to perform a specific task; for example,

```
A> PRINT BOLD.PRN
```

Refer to your printer documentation for more information.

Tips and Traps Summary

Trap: DOS uses CONFIG.SYS only during system startup. Therefore, you must reboot DOS in order for your changes to CONFIG.SYS to take effect.

Creating CONFIG.SYS

Tip: Never modify CONFIG.SYS without first making a backup copy.

Enabling Extended CTRL-BREAK Checking

Tip: To increase the number of times that DOS checks for a CTRL-BREAK, use the BREAK ON command.

Trap: Enabling extended CTRL-BREAK checking increases your system overhead, which makes your application programs run slower.

Tip: To disable extended CTRL-BREAK checking, issue the command BREAK OFF.

Tip: To determine the state of extended CTRL-BREAK processing, simply issue the BREAK command without specifying ON or OFF.

Tip: The CONFIG.SYS BREAK=ON entry enables extended CTRL-BREAK processing at system startup.

Trap: If DOS displays the message

```
Unrecognized command in CONFIG.SYS
```

after you place a BREAK= entry in CONFIG.SYS, you have not specified ON or OFF following the letters "BREAK=."

Tips and Traps Summary (*continued*)

Reducing the Number of Disk I/O Operations

Tip: By increasing the number of disk buffers available in memory, you increase the likelihood of DOS finding the desired information in memory, which reduces the number of slow disk input operations DOS must perform.

Tip: The CONFIG.SYS BUFFERS= entry allows you to define the number of disk buffers available to DOS (2 to 255).

Trap: If you make the number of disk buffers available to DOS too large, you decrease your overall system performance.

Tip: For most applications, 25 disk buffers is sufficient.

Trap: If you turn off your computer or restart DOS while there is data in a disk buffer, the information in the buffer is not recorded to disk and is lost.

Trap: If DOS displays the message

```
Unrecognized command in CONFIG.SYS
```

after you place a BUFFERS= entry in CONFIG.SYS, make sure that the number of buffers you are specifying is in the range from 2 to 255.

Defining Country-Specific Information

Tip: The CONFIG.SYS COUNTRY= entry allows you to select international date, time, and currency formats.

Tips and Traps Summary (*continued*)

Setting Block Device Attributes

Tip: The CONFIG.SYS DRIVPARM= entry allows you to specify the attributes of your block-mode devices.

Trap: If DOS displays the message

```
Unrecognized command in CONFIG.SYS
```

after you include a DRIVPARM= entry in CONFIG.SYS, you have very likely included an invalid value in the list of arguments.

Supporting DOS Applications Prior to Version 2.0

Tip: The CONFIG.SYS FCBS= entry supports older DOS applications that use file control blocks.

Trap: If DOS displays the message

```
Unrecognized command in CONFIG.SYS
```

after you place an FCBS= entry in CONFIG.SYS, you have probably specified an invalid value for the number of FCBs that DOS will support. Specify a value in the range 1 to 255.

Determining the Number of Files That DOS Can Open

Trap: Although DOS provides eight file handles by default, it predefines five of them, restricting your applications to opening only three files at one time.

Tips and Traps Summary (*continued*)

Tip: To increase the number of files that DOS can open at one time, use the CONFIG.SYS FILES= entry.

Trap: If DOS displays the error message

```
Unrecognized command in CONFIG.SYS
```

after you place a FILES= entry in CONFIG.SYS, you have likely specified an invalid number of files. Specify a value in the range 8 to 255.

Defining the Number of Logical Disk Drives Supported

Tip: The CONFIG.SYS LASTDRIVE= entry allows you to specify the letter of the last disk drive that DOS will support.

Trap: DOS will not allow LASTDRIVE to hide existing disk drives.

Tip: To disable the pretender command SUBST, you can use LASTDRIVE to prevent DOS from creating logical disk drives.

Trap: If DOS displays the message

```
Unrecognized command in CONFIG.SYS
```

after you have modified the LASTDRIVE= entry in CONFIG.SYS, make sure that you have specified a disk drive in the range A through Z.

Defining the Command-Line Interpreter

Tip: The SHELL= entry allows you to specify a command interpreter as an alternative to COMMAND.COM.

Tips and Traps Summary (*continued*)

Tip: Fixed-disk users can remove the COMMAND.COM file from the root directory by using the SHELL= entry in CONFIG.SYS.

Trap: If DOS displays the message

```
Bad or missing command interpreter
```

the file name that you have specified for your command interpreter does not exist on disk.

Creating a Hardware Interrupt Stack

Tip: The CONFIG.SYS STACKS= entry allows you to increase the size of the DOS stack. This reduces the likelihood of DOS running out of stack space if successive hardware interrupts occur.

Trap: If DOS displays the message

```
Unrecognized command in CONFIG.SYS
```

after you place a STACKS= entry in CONFIG.SYS, make sure that your stack frame (0-64) and stack size (0-512) values are correct.

Installing a Device Driver

Tip: The CONFIG.SYS DEVICE= entry instructs DOS to install a device driver in memory.

Tip: Most device drivers have the extension SYS.

Tips and Traps Summary (*continued*)

Supporting External Floppy-Disk Drives

Tip: DRIVER.SYS contains the DOS device drivers for external floppy-disk drives.

Trap: If DOS displays the message

```
Unrecognized command in CONFIG.SYS
```

after you include a DRIVER.SYS entry, you have likely specified an invalid parameter in the entry.

Installing a Virtual-Disk Drive in RAM

Trap: Depending on your version of DOS, the RAMDEVICE.SYS device driver may be named VDISK.SYS.

Tip: Once installed, you access your virtual disk with the disk-drive letter displayed in the message

```
Microsoft RAMDrive version 1.19 virtual disk D:
```

Trap: RAM drives lose their contents when DOS restarts.

Tip: If you place data files in your RAM drive, back up the files to a fixed or floppy disk on a regular basis.

Trap: If DOS displays the message

```
Microsoft RAMDrive version 1.19 virtual disk D:
RAMDrive: Invalid parameter
```

when your system restarts, make sure that your RAM-DRIVE.SYS entries are correct.

Tips and Traps Summary (*continued*)

Having Fun with ANSI.SYS

Tip: The ANSI.SYS device driver enhances your screen output and keyboard input capabilities.

Trap: If your screen displays strange numbers or characters when you type a file containing an ANSI escape sequence, either your sequence is incorrect or you do not have ANSI.SYS correctly installed.

10

Utilizing the DOS Environment

Each time DOS boots, it sets aside a 160-byte region in memory called the *DOS environment,* as shown in Figure 10-1. The function of the DOS environment is to provide an area that is available to all of your programs and DOS commands into which you can place information. This chapter examines the DOS environment in detail, working with the DOS SET command, which allows you to place information into and display the contents of the environment. Several DOS commands that place information into the environment as part of their processing are also discussed here.

Displaying the Default Environment

From the DOS prompt, issue the SET command, as shown here:

```
A> SET
```

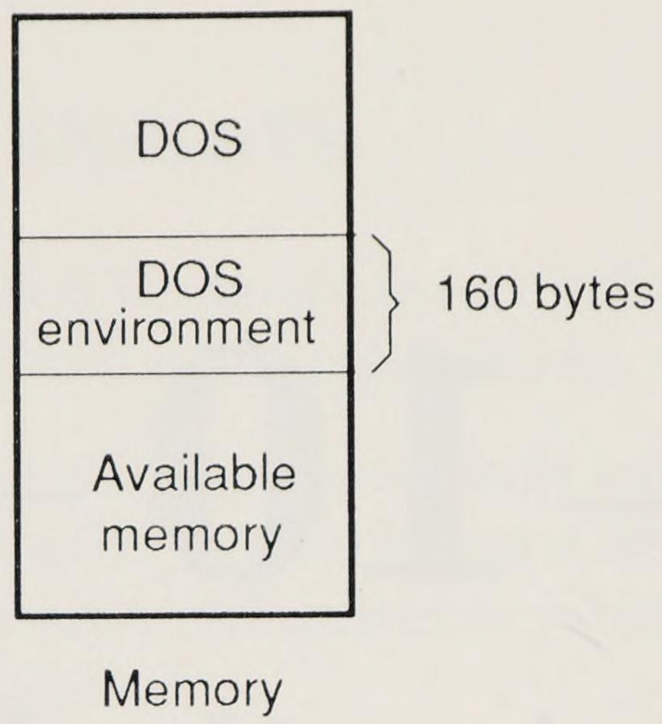


Figure 10-1. The DOS environment

Tip: *SET is an internal DOS command that allows you to place items into the DOS environment or display the current environment entries.*

In this case, since you have not yet used SET to place values into the environment, SET will display the default entries that DOS places there at startup.

```
A> SET
PATH=
COMSPEC=A:\COMMAND.COM
```

Tip: *If you invoke the SET command without command-line parameters, SET will display the contents of the current environment.*

The DOS environment contains single-line entries in the form

```
VARIABLE=VALUE
```

COMSPEC= and PATH= are the two entries that DOS places in the environment when it boots.

```
PATH =
variable

COM SPEC = A:\COMMAND.COM
variable        value
```

The COMSPEC= entry tells DOS the location of the command interpreter. DOS assigns this entry based on the value of the SHELL= entry in CONFIG.SYS. If you don't place a SHELL= entry in CONFIG.SYS, DOS defaults to COMMAND.COM on your boot disk, as discussed in Chapter 9.

Tip: *The COMSPEC= environment entry contains the location of the command interpreter assigned by the SHELL= entry in the CONFIG.SYS file.*

By default, the DOS PATH= entry contains no value but you can assign a value to it by using the DOS SET or PATH command. DOS uses the PATH= entry to locate external commands. As you know, by default, DOS searches for external commands on the current disk or the disk specified in your command line.

```
B> A:DISKCOPY A: B:
```

The PATH= entry simplifies command execution by allowing you to define a list of locations where DOS should look for your external command files if it fails to locate the file in the current directory or specified disk drive. Assume, for example, that your PATH= entry contains

```
PATH=A:;B:
```

If DOS fails to locate a command, it will first search the disk in drive A. If it finds the command on drive A, DOS will execute it. If not, DOS continues its search to drive B. If DOS has not found the file after searching the list of entries associated with PATH, it will display the familiar message

```
Bad command or file name
```

In Chapter 13 DOS subdirectories will be examined and the DOS PATH command—which exists specifically to define the environment PATH= entry—will be discussed. At that time, you will also learn that the DOS APPEND command allows you to assign an environment entry associated with the list of locations that DOS will search for data files (as opposed to command files). For now, remember that DOS always places the COMSPEC= and PATH= entries in the environment.

Assigning Environment Entries

As you have just seen, the DOS SET command allows you to display the current contents of the DOS environment. In a similar manner, SET allows you to place new entries in the environment in the form

```
SET VARIABLE=VALUE
```

Issue the command

```
A> SET CHAPTER=10
```

Then, using SET, display the current environment entries:

```
A> SET
PATH=
COMSPEC=A:\COMMAND.COM
CHAPTER=10
```

As you can see, SET has created an environment entry called CHAPTER, and assigned it the value 10.

Tip: *To assign a variable to the DOS environment, use the SET command in the form SET VARIABLE=VALUE.*

Although the previous environment entry was not useful to DOS, it showed just how easy it is to place values in the DOS environment.

To remove a variable from the environment, you will again enter the SET command.

Tip: *To remove a variable from the DOS environment, issue the SET command in the form SET VARIABLE=.*

When the SET command encounters a variable with no value assigned to it, for example,

```
A> SET CHAPTER=
```

it removes the variable from the environment.

```
A> SET
PATH=
COMSPEC=A:\COMMAND.COM
```

For fun, issue the command

```
A> SET PROMPT=YES$g
```

DOS will respond to the SET command by changing your system prompt, as shown here:

```
A> SET PROMPT=YES$g
YES>
```

If you issue the SET command

```
YES> SET
```

DOS will display your current environment entries:

```
YES> SET
PATH=
COMSPEC=A:\COMMAND.COM
PROMPT=YES$g
```

As you can see, the last environment entry defines your system prompt. As it turns out, each time DOS completes a command, it searches your environment for an entry in the form

```
PROMPT=
```

If DOS finds this entry, it sets your system prompt as you specified; otherwise, it simply displays the familiar

```
A>
```

prompt. To reset your prompt to the drive letter, remove the PROMPT variable:

```
YES> SET PROMPT=
A>
```

DOS restores the prompt since it no longer finds the PROMPT= environment entry. In Chapter 18, several DOS commands (such as PROMPT) that affect the entries in the DOS environment will be discussed.

How Many Entries Can The Environment Store?

Remember that DOS initially creates an environment of 160 bytes. As long as you don't install any memory-resident software—such as PRINT, GRAPHICS, or GRAFTABL—DOS allows your environment to grow to up to 32,768 bytes (see Figure 10-2). Once you install memory-resident software, however, you restrict the growth of your environment, as in Figure 10-3.

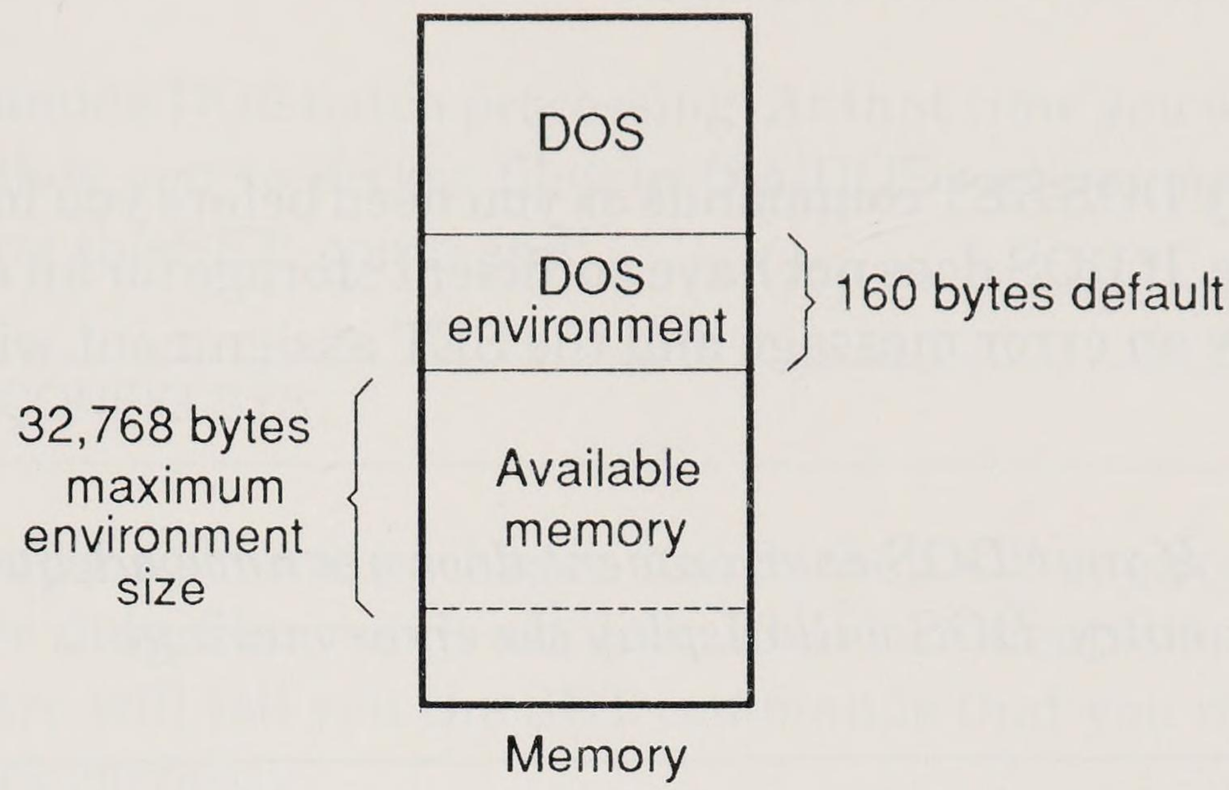


Figure 10-2. The DOS environment can grow to 32,768 bytes

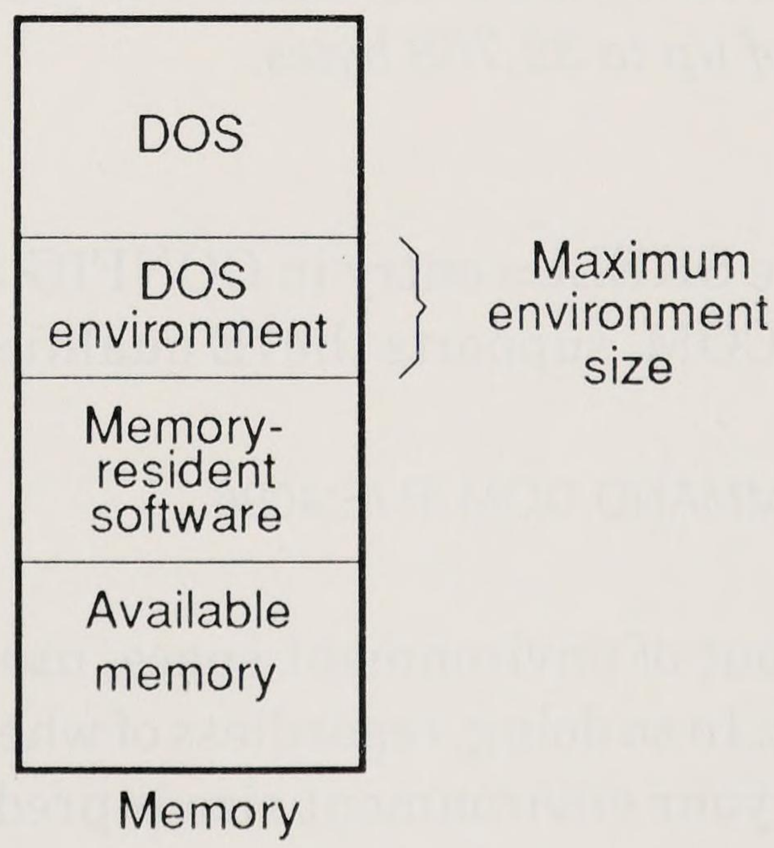


Figure 10-3. Memory-resident software restricts DOS environment growth

Trap: *Once you install memory-resident software, you restrict the size of the DOS environment.*

Issue as many DOS SET commands as you need before you install memory-resident software. If DOS does not have sufficient storage for an environment entry, it will display an error message and the SET assignment will fail.

Trap: *If your DOS environment does not have adequate space for a new entry, DOS will display the error message*

```
Out of environment space
```

and your SET command will fail.

This error message is not restricted to the DOS SET command. As discussed, the PATH, APPEND, and PROMPT commands also use the DOS environment. If your environment is full, these commands will fail.

Tip: *The CONFIG.SYS SHELL= entry allows you to define an environment of up to 32,768 bytes.*

Remember that the SHELL= entry in CONFIG.SYS, when used in conjunction with COMMAND.COM, supports the /E qualifier to define environment size:

```
SHELL=A:\COMMAND.COM /P /E:4096
```

If DOS runs out of environment space, use this entry to increase the size of the environment. In so doing, regardless of whether or not you install memory-resident software, your environment size is predefined.

Other Uses of the DOS Environment

Chapter 11 will examine DOS batch processing. At that time you will learn that DOS batch files allow you to access files in the DOS environment as named parameters by using the SET command:

```
A> SET FILE=CONFIG.SYS
```

Many third-party software packages search the DOS environment for information on where data files reside on disk. The documentation that accompanies such software will tell you the SET commands that you must issue for the software to work correctly.

The PATH, APPEND, and PROMPT commands will be discussed in Chapter 18, where you will learn how each affects your environment.

Tips and Traps Summary

Displaying the Default Environment

Tip: SET is an internal DOS command that allows you to place items into the DOS environment or display the current environment entries.

Tip: If you invoke the SET command without command-line parameters, SET will display the contents of the current environment.

Tip: The COMSPEC= environment entry contains the location of the command interpreter assigned by the SHELL= entry in CONFIG.SYS.

Assigning Environment Entries

Tip: To assign a variable to the DOS environment, use the SET command in the form SET VARIABLE=VALUE.

Tip: To remove a variable from the DOS environment, issue the SET command in the form SET VARIABLE=.

How Many Entries Can the Environment Store?

Trap: Once you install memory-resident software, you restrict the size of the DOS environment.

Trap: If your DOS environment does not have adequate space for a new entry, DOS will display the error message

```
Out of environment space
```

and your SET command will fail.

Tips and Traps Summary (*continued*)

Tip: The CONFIG.SYS SHELL= entry allows you to define an environment of up to 32,768 bytes.

11

Using DOS Batch Files

So far, all of the commands examined in this book have been *interactive* commands; that is, you issue a command, wait for it to complete, and then issue the next command. Depending on the commands you are running, executing commands in this manner can tie up a considerable amount of your time at the keyboard, when your goal is to execute subsequent commands.

To simplify command execution, DOS allows you to group commands into standard text files (called *batch files*) that have the extension *BAT*. In some cases you will use DOS batch files to abbreviate long commands. At other times, you will group logically related commands into one batch file. Regardless of your intent, a batch file is nothing more than a text file containing one or more commands that DOS is to execute.

Tip: *DOS batch files allow you to abbreviate a long command, or group logically related commands into a standard text file, with the extension BAT.*

A First Look at Batch Files

As stated, a batch file is a text file that contains one or more DOS commands. First, create a batch file called DT.BAT that allows you to issue the DOS DATE and TIME commands with one command:

```
A> COPY CON B:DT.BAT
DATE
TIME
^Z
        1 File(s) copied
```

In this case, the batch-file name is DT.BAT (note the BAT extension). The two DOS commands in the batch file are DATE and TIME. To execute these commands, type the batch-file name at the DOS prompt, as shown here:

```
A> B:DT
```

Tip: *To execute a DOS batch file, type the batch-file name at the DOS prompt and press ENTER.*

DOS will respond by executing the first command in the batch file, which in this case is the DOS DATE command.

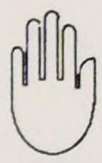
```
A> DATE
Current date is Mon 6-27-1988
Enter new date (mm-dd-yy):
```

At this point, executing the batch file is no different than executing the DOS DATE command. You can type in a new system date or press ENTER to leave the date unchanged. In either case, as soon as you complete the DOS DATE command, DOS will execute the TIME command.

```
A> DATE
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy):

A> TIME
Current time is 15:09:20.73
Enter new time:
```

As you can see, unless it is told otherwise (by a GOTO or IF command, discussed in Chapter 12), DOS always executes the commands in your batch file from top to bottom in the order you specify. Once you complete the DOS TIME command, no more commands remain in the batch file and DOS displays its prompt.

```
A> DATE
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy):

A> TIME
Current time is 15:09:20.73
Enter new time:

A>
```



Trap: *If DOS encounters an invalid command within your batch file, it will display the message*

```
Bad command or file name
```

and continue processing with the next command in the file.

If your DOS batch file contains

```
DATE
TIME
```

DOS will display

```
A> B:DT
Bad command or file name

A> TIME
Current time is 15:11:54.41
Enter new time:
```

when you invoke it. As you can see, DOS ignored the invalid command, continuing with the next command in the batch file.

The following batch file clears your screen, displays the volume label of the current disk, and then performs a directory listing:

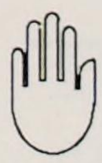
```
A> COPY CON B:DISKINFO.BAT
CLS
VOL
DIR
^Z
        1 File(s) copied
```

To execute this batch file, enter the batch-file name, as shown here:

```
A> DISKINFO
```

Tip: *Make your batch-file names as meaningful as possible.*

Just as you should name all of your DOS files in a meaningful fashion, DOS batch files are no exception. Your batch-file name should remind you of the processing that it performs.



Trap: *If you give a batch file the same name as a DOS internal or external command, DOS will not execute your batch file.*

Each time you type a DOS command, DOS first tests to see if the command is an internal DOS command. If not, it checks whether the command is an external command. If DOS has still not found the command, it searches for a batch file with the given name. If you create a batch file with the name CLS.BAT, DOS will never find it, but will instead always execute the internal CLS command. Figure 11-1 illustrates this process.

Terminating a Batch File

Just as there are many times when you want to cancel a DOS command before it completes, there will be times when you want to terminate a DOS batch file.

Tip: *The CTRL-BREAK key combination allows you to terminate a DOS batch file.*

Issue the DT.BAT file that you created earlier in this chapter.

```
A> B:DT
```

When DOS displays the DATE prompt, press CTRL-BREAK. DOS will respond with

```
A> DATE
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy): ^C

Terminate batch job (Y/N)?
```

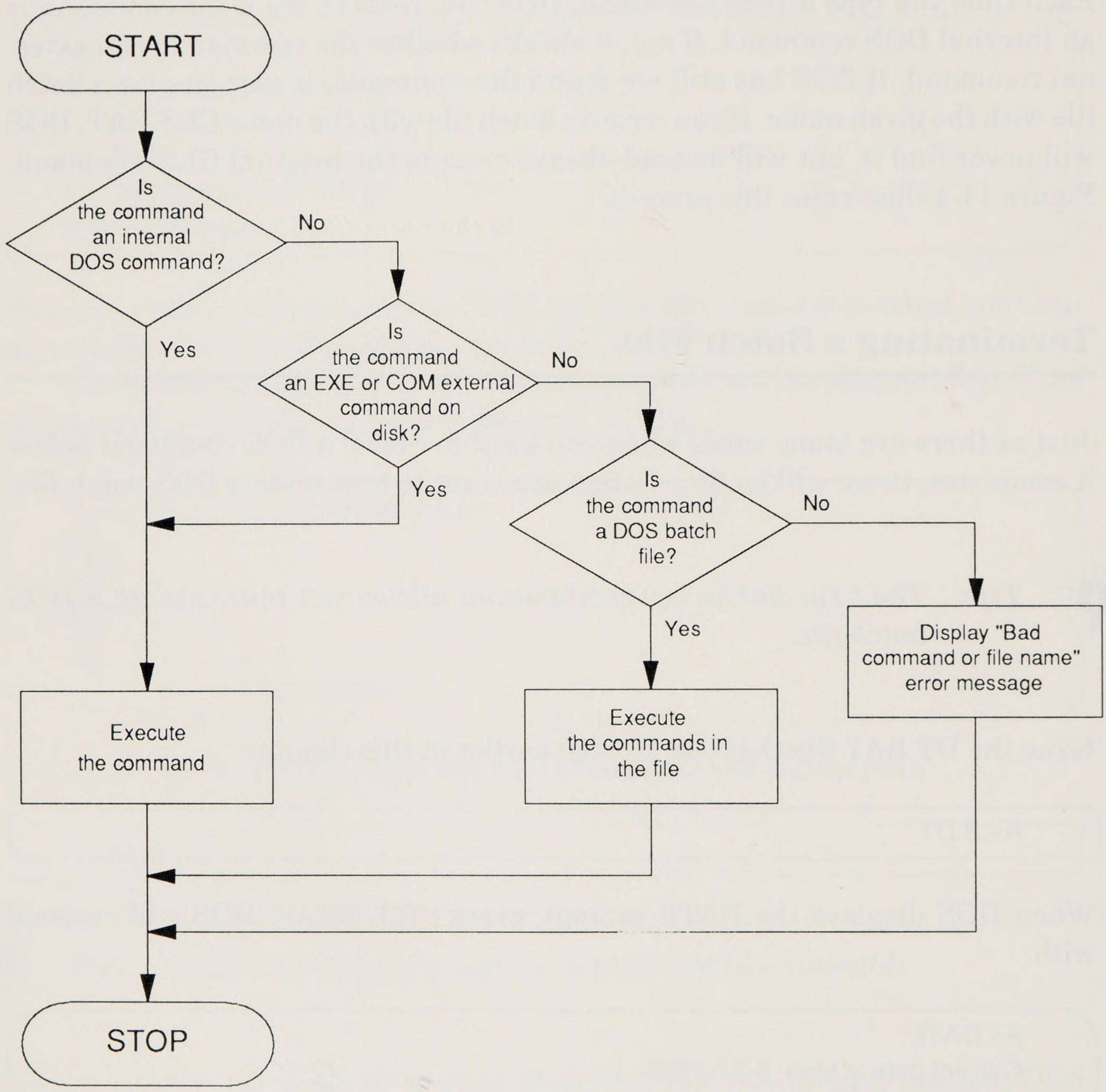


Figure 11-1. Process of DOS executing commands

This prompt allows you to terminate either the command that was previously executing or the entire batch file.

Tip: *If you respond with* **Y** *to the prompt*

```
Terminate batch job (Y/N)?
```

DOS will terminate the current batch file. If instead, you type **N**, *then DOS will terminate only the previously executing command, continuing execution of the batch file.*

In this case, type **Y** and press ENTER. DOS will terminate the current batch file, redisplaying the DOS prompt:

```
A> DATE
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy): ^C

Terminate batch job (Y/N)? Y

A>
```

Again, invoke this batch procedure, pressing CTRL-BREAK at the DATE prompt.

```
A> DATE
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy): ^C

Terminate batch job (Y/N)?
```

This time, respond to the prompt with **N**.

```
A> DATE
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy): ^C

Terminate batch job (Y/N)? N
```

As you can see, DOS terminates the DATE command, continuing with the next command in the batch file.

```
A> DATE
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy): ^C

Terminate batch job (Y/N)? N

A> TIME
Current time is 15:09:20.73
Enter new time:

A>
```

Suppressing Command-Name Display

Both of the previous batch files displayed the name of each command on the screen as the command executed.

```
A> B:DT

A> DATE
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy):

A> TIME
Current time is 15:09:20.73
Enter new time:
```

In some cases, however, you won't want DOS to display command names on the screen. The DOS ECHO command allows you to suppress the command names from being displayed.

Tip: *The ECHO OFF command suppresses command-name display in batch-file processing.*

Modifying the DT.BAT file to support the ECHO OFF command produces

```
A> COPY  CON  B:DT.BAT
ECHO OFF
DATE
TIME
^Z
            1 File(s) copied
```

When you execute this procedure, DOS will display

```
A> B:DT

A> ECHO  OFF
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy):
Current time is 15:09:20.73
Enter new time:
```

Tip: *DOS 3.3 allows you to suppress batch-file command names by preceding the command name with the @ character.*

For example, enter

```
A> COPY  CON  B:DT.BAT
@DATE
@TIME
^Z
            1 File(s) copied
```

When you later execute this procedure, DOS will display

```
A> B:DT
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy):
Current time is 15:09:20.73
Enter new time:
```

Most DOS version 3.3 users reduce the number of @ characters in their batch files by preceding the ECHO OFF command with @, as shown here:

```
A> COPY CON B:DISKINFO.BAT
@ECHO OFF
CLS
VOL
DIR
^Z
        1 File(s) copied
```

Tip: *In addition to suppressing command-name display in your batch files, the ECHO command allows you to display messages for the user.*

To direct ECHO to display a message, type the desired message on the ECHO command line, as shown here:

```
ECHO This is a message
```

The following DOS batch file shows the user a valid date and time immediately before the DOS DATE and TIME commands:

```
A> COPY CON B:DT.BAT
ECHO OFF
ECHO Sample Date (Jun 11, 1988) is 6/11/88
DATE
ECHO Sample Time (12:30 PM) is 12:30
TIME
```

When you invoke this file, it will display

```
A> B:DT
Sample Date (Jun 11, 1988) is 6/11/88
Current date is Mon  6-27-1988
Enter new date (mm-dd-yy):
Sample Time (12:30 PM) is 12:30
Current time is 15:09:20.73
Enter new time:
```

Many batch files use ECHO to display a series of instructions, or copyright information, as shown here:

```
ECHO OFF
ECHO ******************************************
ECHO Place a newly formatted diskette in drive B
ECHO ******************************************
ECHO Make sure that your printer is turned on and
ECHO the on-line light is illuminated.
ECHO Ensure that your printer has the company
ECHO checks ready for printing.
ECHO ******************************************
```

Tip: *If you have installed the ANSI.SYS device driver as discussed in Chapter 9, you can use the ECHO command to enable various screen colors and key definitions.*

Using EDLIN, the following batch file sets your screen background color to blue and then displays a directory listing. Once the directory listing completes, the screen colors are returned to their defaults.

```
A> EDLIN  B:BLUEDIR.BAT
New file
*I
```

```
                1: ECHO OFF
                2: ECHO ^V[[44m
                3: DIR
                4: ECHO ^V[[0m
                5: ^C
*E
```

If you are using a monochrome system, you might consider using the escape sequence ESC[1m, which selects bold text as opposed to the background color of blue.

Documenting Your Batch Files

Although your batch-file names should give a good idea of the processing that your batch file is performing, there may still be times when you won't remember the processing that is occurring within the batch file. The DOS REM command allows you to place REMarks (or *comments*) within your batch files to explain the processing. REM does not change the batch file's execution. It simply allows you to include a series of remarks that you can refer back to if you ever need to change the batch file or determine its processing.

Consider how remarks improve the readability of the previous batch file, which sets your screen's background color to blue.

```
A> EDLIN  B:BLUEDIR.BAT
End of input file
*L
                1:  ECHO OFF
                2:  REM Set the screen background color to blue using the
                3:  REM ANSI escape sequence Esc[nm
                4:  ECHO ^V[[44m
                5:  REM Display the directory listing in blue
                6:  DIR
                7:  REM Set the screen background color back to black
                8:  ECHO ^V[[0m
*
```

In this case, each step of the processing is quite well documented.

Tip: *The REM command allows you to place meaningful remarks in a batch file to explain the file's processing.*

If you spend a few extra minutes after you create your batch file to ensure that the file is well documented, you will save a considerable amount of time later if you ever have to determine the file's actual processing.

Trap: *If your batch file does not contain the command ECHO OFF, DOS will display all of the REM statements that are in the file.*

When you invoke the follow batch file

```
A> COPY  CON  B:REMTEST.BAT
REM This batch file displays the current volume label, and then
REM issues the DOS LABEL command.
VOL
LABEL
^Z
                1 File(s) copied
```

DOS will display

```
A> B:REMTEST

A> REM This batch file displays the current volume label, and then

A> REM issues the DOS LABEL command.

A> VOL

Volume in drive A is MS330PP01

A> LABEL

Volume in drive A is MS330PP01

Volume label (11 characters, ENTER for none)?
```

To disable display of your remarks, include the ECHO OFF command, as shown here:

```
ECHO OFF
REM This batch file displays the current volume label, and then
REM issues the DOS LABEL command.
VOL
LABEL
```

When you invoke this batch file, DOS will no longer display the lines containing the REM commands.

Suspending Batch Processing

Depending on your batch processing requirements, you may have to suspend processing long enough for users to insert a new disk in a drive, or to ensure that the printer is on-line.

Tip: *The PAUSE command allows you to temporarily suspend batch processing, displaying an optional message until the user presses any key to continue.*

The format of the DOS PAUSE command is

```
PAUSE message
```

The following batch file, for example, uses PAUSE to suspend processing until the user turns on the printer and presses a key to continue:

```
PAUSE Make sure that your printer is on-line
COPY EXPENSES.JUN PRN:
```

When this file is invoked, it will display

```
PAUSE Make sure that your printer is on-line
Strike a key when ready . . .
```

Trap: *The ECHO OFF command suppresses display of messages from the PAUSE command.*

In this batch file, for example, the ECHO OFF command will suppress the messages normally displayed by the PAUSE command:

```
ECHO OFF
PAUSE Make sure that your printer is on-line
COPY EXPENSES.JUN PRN:
```

Invoking this batch file results in

```
A> ECHO  OFF
Strike a key when ready . . .
```

Batch-File Parameters

Although DOS batch files can often simplify command processing, there are many cases when you can increase the number of applications that a batch file supports by using DOS batch parameters.

Consider this batch file, which uses the ANSI.SYS driver to set your screen's background color to blue:

```
A> EDLIN  B:BLUESCR.BAT
New file
*I
          1:* ECHO ^V[[44m
          2:*^C
*E

A>
```

When you execute this batch file, DOS will set your screen color as desired. Chapter 9 discussed the fact that the ANSI.SYS device driver supports many different colors, as shown in Table 11-1.

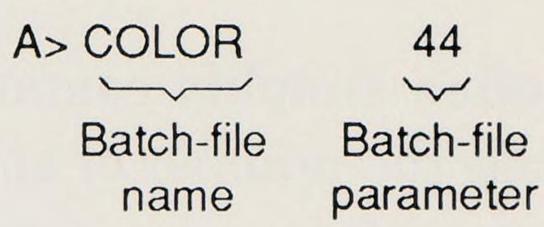
If you are instead using a monochrome system, create the BOLDSCR.BAT file, which sets your screen character attributes to bold display:

```
A> EDLIN  B:BOLDSCR.BAT
New file
*I
            1:*ECHO ^V[[lm
            2:*^C
*E
A>
```

A more useful batch file for users with color monitors would set the screen to any user-specified color, not just to blue. To select the background color blue, for example, you could enter the command

```
A> COLOR  44
```

In this case, COLOR is the name of a DOS batch file, and 44 is a batch-file parameter.



Tip: *Batch-file parameters increase the usefulness of your batch files.*

Starting with a simple example, this batch file uses the DOS ECHO command to display the first command parameter:

```
A> COPY  CON  B:ECHOTEST.BAT
ECHO %1
^Z
            1 File(s) copied
```

Table 11-1. ANSI.SYS screen-color settings

Value	Function
0	Default attributes
1	Bold text
2	Low-intensity text
3	Italic text
5	Blinking text
6	Rapid blinking text
7	Reverse video on
8	Concealed text
30	Black foreground
31	Red foreground
32	Green foreground
33	Yellow foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground
40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background
48	Subscript
49	Superscript

If you invoke this batch file as

```
A> B:ECHOTEST A
```

DOS will display

```
A> B:ECHOTEST A

A> ECHO A

A

A>
```

Likewise, if you invoke the batch file with multiple entries in the command line

```
A> B:ECHOTEST A B C
```

the batch file still displays only the first command-line parameter:

```
A> B:ECHOTEST A

A> ECHO A

A

A>
```

Tip: *DOS uses the space character to separate command parameters.*

Since the previous command line contained spaces between the letters A, B, and C, DOS assigns command-line parameters, as shown here:

```
A> B: ECHOTEST  A  B  C
      Batch file  %1 %2 %3
```

By default, DOS allows you to reference nine batch parameters, using %1 to %9. This batch file displays nine batch-file parameters:

```
A> COPY  CON  B:ECHOTEST.BAT
ECHO OFF
ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9
^Z
            1 File(s) copied
```

If you invoke the batch file with more than nine parameters, DOS still displays only nine:

```
A> B:ECHOTEST  A  B  C  D  E  F  G  H  I  J  K  L  M

A> ECHO  OFF
A B C D E F G H I

A>
```

Likewise, less than nine parameters is also fully supported:

```
A> B:ECHOTEST  A  B  C  D  E

A> ECHO  OFF
A B C D E

A>
```

Tip: *DOS assigns the name of the batch file to %0.*

Using the fact that DOS assigns the batch-file name to %0, this batch procedure displays its name on one line, and then the batch-file parameters on the following line:

```
A> COPY  B:ZEROTEST.BAT
ECHO OFF
ECHO %0
```

```
ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9
^Z
        1 File(s) copied
```

Upon invocation, this batch file will display

```
A> B:ZEROTEST A B C

A> ECHO OFF
B:ZEROTEST
A B C

A>
```

With your knowledge of batch-file parameters, examine another batch file, which sets your screen's background color. In this case, the user can specify the desired color as a batch parameter.

```
A> EDLIN B:COLOR.BAT
New file
*I
        1:* ECHO ^V[[%1m
        2:* ^C
*E

A>
```

The DOS ECHO command uses the escape sequence Esc[*n*m to set the screen color. In this case, ECHO will first write the characters "Esc[." Next, it will write the number specified by the user in %1, followed by the letter "m," which terminates the escape sequence.

As such, the command

```
A> B:COLOR 1
```

sets your screen to bold text display.

In a similar manner, you can create a batch file that allows the user to easily assign a value to the F10 function key. In this case, the file name is F10.BAT.

```
A> B:F10 DIR
```

This batch file uses the ANSI.SYS device driver to define the F10 key, as discussed in Chapter 9:

```
A> EDLIN  B:F10.BAT
New File
*I
          1:ECHO ^V[[0;68;"%1"p
          2:^C
*E

A> B:F10  DIR

A>
```

In Chapter 12, you will learn how to improve your batch-file processing by invoking a batch file with

```
A> COLOR BLUE
```

Your batch file will determine the correct processing required to set the screen color.

Tips and Traps Summary

Tip: DOS batch files allow you to abbreviate a long command or group logically related commands into a standard text file with the extension BAT.

A First Look at Batch Files

Tip: To execute a DOS batch file, type the batch-file name at the DOS prompt and press ENTER.

Trap: If DOS encounters an invalid command within your batch file, it will display the message

```
Bad command or file name
```

and continue processing with the next command in the file.

Tip: Make your batch-file names as meaningful as possible.

Trap: If you give a batch file the same name as a DOS internal or external command, DOS will not execute your batch file.

Terminating a Batch File

Tip: The CTRL-BREAK key combination allows you to terminate a DOS batch file.

Tip: If you respond with **Y** to the prompt

```
Terminate batch job (Y/N)?
```

DOS will terminate the current batch file. If, instead, you type **N**, DOS will terminate only the previously executing command, continuing execution of the batch file.

Tips and Traps Summary (*continued*)

Suppressing Command-Name Display

Tip: The ECHO OFF command suppresses command-name display in batch-file processing.

Tip: DOS 3.3 allows you to suppress batch-file command names by preceding the command name with the @ character.

Tip: In addition to suppressing command-name display in your batch files, the ECHO command allows you to display messages for the user.

Tip: If you have installed the ANSI.SYS device driver as discussed in Chapter 9, you can use the ECHO command to enable various screen colors and key definitions.

Documenting Your Batch Files

Tip: The REM command allows you to place meaningful remarks in a batch file to explain the file's processing.

Trap: If your batch file does not contain the command ECHO OFF, DOS will display all of the REM statements that are in the file.

Suspending Batch Processing

Tip: The PAUSE command allows you to temporarily suspend batch processing, displaying an optional message until the user presses any key to continue.

Trap: The ECHO OFF command suppresses display of messages from the PAUSE command.

Tips and Traps Summary (*continued*)

Batch-File Parameters

Tip: Batch-file parameters increase the usefulness of your batch files.

Tip: DOS uses the space character to separate command parameters.

Tip: DOS assigns the name of the batch file to %0.

12

Building Intelligent Batch Files

In Chapter 11 you started working with DOS batch files. At that time, you found out that unless DOS is told otherwise, it always executes batch-file commands sequentially, from top to bottom.

```
CLS
DATE
TIME
```

In this chapter you'll learn how to create "intelligent" batch files that perform decision-making, allow you to repeat a series of commands until a specific condition is met, and allow you to execute a specific command in conjunction with a given set of files. Chapter 11 got you started with your batch files; this chapter teaches you how to fully exploit them.

Using IF to Make Decisions

As just discussed, unless DOS is told otherwise, batch files execute their commands from top to bottom sequentially. However, the DOS IF command allows you to control which commands your batch file will execute, based on one of three conditions. The first, and easiest condition to use is

```
IF EXIST FILENAME.EXT DOS_command
```

This condition directs DOS to perform the command that is specified by DOS_command only if the given file name resides on disk. For example, the command

```
IF EXIST CONFIG.SYS TYPE CONFIG.SYS
```

directs DOS to display the contents of the CONFIG.SYS file if it is found on disk. In a similar manner, the command

```
IF EXIST B:NUMBERS.DAT DEL A:NUMBERS.DAT
```

searches for the NUMBERS.DAT file on drive B, and if the file is found, deletes it from drive A.

Tip: *The IF EXIST FILE condition directs DOS to execute a command only if the specified file resides on disk.*

The second IF condition directs DOS to perform a command only if two *strings* are equal. A string is a group of consecutive characters.

```
IF STRING1==STRING2 DOS_command
```

In this case, the command

```
A> EDLIN  B:COLOR.BAT
New file
*I
            1:* IF %1==BLUE ECHO^[V[44m
            2:* ^C
*E
```

tests to see if the first batch parameter is the word BLUE. If so

```
A> B:COLOR BLUE
```

the IF command will evaluate as true and DOS will use the ANSI.SYS device driver to set your screen's background color to blue.

Tip: *The string1==string2 condition directs DOS to perform the specified command only when the characters in the first sequence match the characters in the second sequence exactly.*

This batch file enhances your previous processing, allowing you to set your screen's background color to one of several colors based on the first batch parameter (%1):

```
A> EDLIN  B:COLOR.BAT
New file
*I
            1:* ECHO OFF
            2:* IF %1==RED ECHO ^V[[41m
            3:* IF %1==GREEN ECHO ^V[[42m
            4:* IF %1==ORANGE ECHO ^V[[43m
            5:* IF %1==BLUE ECHO ^V[[44m
            6:* ^C
*E

A>
```

Trap: *For the IF condition string1==string2 to evaluate as true, the characters in both strings must match in terms of case.*

If your batch file contains the line

```
IF %1==BLUE DOS_command
```

and you invoke the batch file with

```
A> B:COLOR Blue
```

DOS will evaluate the condition

```
IF Blue==BLUE DOS_command
```

as false, since the letters differ in case.

The last IF condition is based on the success of the previous command in the batch file. Many DOS commands provide *exit status values.* These values specify whether the command completed, and if not, why not. For example, the DOS FORMAT command provides the following exit status values:

Value	Meaning
0	Successful completion
3	User-terminated command with CTRL-BREAK
4	Fatal disk error
5	User-terminated at the prompt "Proceed with format (Y/N)?"

If you place the FORMAT command in a batch file, you can test its success by using the DOS IF ERRORLEVEL directive.

Tip: *The IF ERRORLEVEL condition instructs DOS to test the success level of the previous command. As long as the program terminates with a value that is greater than or equal to the value you specify, DOS will execute the given command.*

In the batch file

```
FORMAT
IF ERRORLEVEL 3 DOS_command
```

when the FORMAT command completes, DOS will examine its exit status value. If that value *is greater than or equal* to 3, DOS will execute the second command. In later chapters you will use this form of the DOS IF command in conjunction with the DOS BACKUP and RESTORE commands.

Using the DOS NOT Operator

In many cases, you will want your batch files to perform a specific operation when a given condition fails. The DOS NOT operator provides this capability.

Tip: *The DOS NOT operator instructs DOS to perform the specified command in an IF command when the condition tested fails.*

The following command, for example, instructs DOS to copy the contents of the CONFIG.SYS file to drive B only if that file does not already reside on that disk:

```
IF NOT EXIST B:CONFIG.SYS COPY A:CONFIG.SYS B:
```

As your batch processing becomes more complex, you will use the DOS NOT operator on a regular basis.

Branching Within Your Batch Files Using GOTO

The DOS GOTO command allows you to skip one or more commands or repeat a series of commands, depending on your batch processing requirements. The format of the DOS GOTO command is

GOTO label

where *label* is a named location in your batch file.

Tip: *The GOTO command directs DOS to skip one or more command or to repeat a series of DOS commands.*

DOS *label names* appear in your batch file preceded by a colon, as shown here:

```
GOTO SKIP
DATE
TIME
:SKIP
VER
```

In this case, SKIP is the name of the label that defines the location within the batch file where DOS should resume processing. If you invoke this batch file from the DOS prompt, DOS will begin by executing the first command in the file, which in this case is GOTO SKIP. It will bypass the DATE and TIME commands, resuming execution at the DOS label command, as shown here:

```
GOTO SKIP
DATE
TIME
: SKIP
VER
```

When DOS encounters the colon in the label SKIP, it knows that the line contains a label. Therefore, DOS ignores the contents of the line.

Tip: *DOS batch labels appear in a batch file preceded by a colon. When DOS encounters the label, it ignores the line, continuing with the next command in the batch file.*

When you invoke the previous batch file, for example, DOS will skip the DATE and TIME commands, displaying the DOS version number as directed by the DOS VER command:

```
A> B:GOTOTEST

A> GOTO  SKIP

A> VER

IBM Personal Computer DOS Version  3.30

A>
```

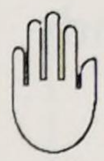
In a similar manner, the following batch file uses the DOS GOTO command to display a continual directory listing of your disk until you press CTRL-BREAK (or ^C) to terminate the command:

```
:REPEAT
DIR
GOTO REPEAT
```

When DOS executes this procedure, it encounters the label REPEAT, which it ignores, continuing with the DIR command. Once the directory listing completes, DOS executes the GOTO command, which branches control back to the label REPEAT.

```
: REPEAT
  DIR
  GOTO REPEAT
```

At that time, DOS repeats the same process. This series of commands will repeat indefinitely.



Trap: *If DOS cannot find the label specified in a GOTO command within the batch file, it will display the error message*

```
Label not found
```

and terminate the batch-file processing immediately.

In this case, the label DONE does not exist in the batch file:

```
A> COPY  CON  B:BADLABEL.BAT
GOTO DONE
DATE
TIME
DOEN
VER
^Z
                1 File(s) copied
```

Therefore, when you execute this batch file, DOS will display

```
Label not found
```

Trap: *DOS distinguishes between only the first eight characters of your label names.*

Although your batch label names can be virtually any length, DOS uses only the first eight characters of the label name. If you have the labels

```
DONE_PART1

DONE_PART2
```

DOS will not differentiate between the two, assuming that each references the eight-character label DONE_PAR.

The following batch file demonstrates this fact, using the first command-line parameter. The intent of the batch file is to display the message

```
Value entered was 1
```

when the user enters the command as

```
A> B:VALUE 1
```

and the message

```
Value entered was 2
```

when the user enters the command

```
A> B:VALUE 2
```

However, because the batch file uses the label names VALUE_WAS1 and VALUE_WAS2, DOS assumes that both GOTO commands reference the label VALUE_WA.

```
A> COPY CON B:VALUE.BAT
ECHO OFF
IF %1==1 GOTO VALUE_WAS1
IF %2==2 GOTO VALUE_WAS2
GOTO DONE
:VALUE_WAS1
ECHO Value entered was 1
GOTO DONE
:VALUE_WAS2
ECHO Value entered was 2
:DONE
^Z
            1 File(s) copied
```

Make your label names as meaningful as possible, while keeping in mind that DOS examines only the first eight characters of the name.

The following batch file demonstrates how you can use the DOS GOTO command in conjunction with the DOS IF command:

```
IF NOT EXIST B:CONFIG.SYS GOTO DO_COPY
PAUSE CONFIG.SYS Already exists on disk in drive B
DO_COPY
COPY A:CONFIG.SYS B:
```

In this case, if the CONFIG.SYS file already exists on the disk in drive B, the batch file will display the message

```
A> PAUSE  CONFIG.SYS Already exists on disk in drive B
Strike a key when ready . . .
```

If you don't want to overwrite the file, press CTRL-BREAK to terminate the batch file, avoiding the file copy.

Testing for a Nonexistent Parameter

As you just saw, by using the DOS IF command to compare two character strings, you can invoke a batch file using

```
A> B:COLOR  BLUE
```

and the batch file will test the contents of %1 to determine the corresponding processing. In such cases, you may have to ensure that the user specifies at least one batch parameter. When a user does not specify one of the batch parameters (%1 to %9), DOS assigns an *empty* or *null string* to the parameter. To test for the empty string, use the command

```
IF '%1'==" ECHO NO PARAMETER
```

If the user omits the %1 parameter,

```
A> B:COLOR
```

the test will become

```
IF "==" ECHO NO PARAMETER
```

Later sections of this chapter will use this test in conjunction with the DOS SHIFT command.

Tip: *To test for a nonexistent batch parameter, use the condition IF '%1'==". If you do not specify a batch parameter, the condition becomes IF "==".*

Applying a DOS Command To a Collection of Files

Just as the DOS GOTO command allows you to repeat a series of commands until the user presses CTRL-BREAK to terminate the batch file, the DOS FOR command allows you to repeat a specific DOS command for a given set of files.

Tip: *The FOR command instructs DOS to repeat a specific command for a given set of DOS files.*

Assume, for example, that each month you track your monthly expenses in files named MONTH.EXP, as shown here:

```
A> DIR

Volume in drive A has no label
Directory of  A:\

JAN          EXP          2816 10-03-84 10:33p
FEB          EXP          2816 10-03-84 10:33p
MAR          EXP          2816 10-03-84 10:33p
          3 File(s) 353280 bytes free
```

As you know, the DOS TYPE command does not support the DOS wildcard characters. Therefore, if you issue the command

```
A> TYPE *.EXP
```

the command will fail and DOS will display

```
Invalid filename or file not found
```

If you want to display all of your monthly expense files at once, you can create a batch file that contains several single-line TYPE commands, as shown here:

```
TYPE JAN.EXP
TYPE FEB.EXP
TYPE MAR.EXP
```

At the end of each month, you add the file that contains the month's expenses to the batch file. When you issue this batch file, DOS will display each of your expense files. The DOS FOR command, however, provides you with a processing alternative.

In this case, the batch file contains

```
FOR %%F IN (JAN.EXP FEB.EXP MAR.EXP) DO TYPE %%F
```

Examining this command, the set of files is easily identified.

```
For %%F IN (JAN.EXP FEB.EXP MAR.EXP) DO TYPE %%F
            \_____________________/   \_____/
                     Files              DOS
                                      command
```

As before, this batch file is issuing the DOS TYPE command. However, the %%F characters require additional explanation. When the DOS FOR command executes, it assigns the first file in the set of files to a variable with a name in the form *%%name*. In this case, the variable name is %%F. In the command

```
FOR %%F IN (JAN.EXP FEB.EXP MAR.EXP) DO TYPE %%F
```

FOR first substitutes the file name JAN.EXP for %%F. When DOS issues the command TYPE %%F, the substitution results in TYPE JAN.EXP. As soon as the TYPE command completes, DOS assigns the next file in the set of files to %%F. Next, it issues the command TYPE FEB.EXP.

When the TYPE command completes, FOR repeats this cycle, displaying the contents of MAR.EXP. When FOR next checks the set of files for an additional file, none remain, and the FOR command completes.

Although this processing is more efficient than the previous batch file, which used several lines of DOS TYPE commands, it still doesn't eliminate the fact that each month you have to edit the batch file to add that month's file to the set of files to display.

Tip: *The FOR command supports DOS wildcard characters.*

If you place DOS wildcard characters in the list of files FOR is to process, for example

```
FOR %%F IN (*.EXP) DO TYPE %%F
```

FOR will expand on the wildcard character, displaying all of the files with the extension EXP in the same order in which the files appear in a directory listing. Although this batch file better suits monthly expense processing, you can enhance it to suit many more applications if you base the processing on batch parameters, as shown here:

```
FOR %%F IN (%1) DO TYPE %%F
```

In this case, assuming that you have named the file T.BAT, you can issue such commands as

```
A> T CONFIG.SYS
A> T *.EXP
A> T *.*
```

As you can see, by combining the DOS FOR command with batch parameters, your processing is essentially unrestricted. Later in this chapter you will again modify this batch procedure, increasing its functionality even more.

Tip: *Although FOR is most commonly used in DOS batch files, DOS also allows you to invoke FOR from the DOS prompt.*

You just found that the format of FOR within batch files is

```
FOR %%variable IN (set) DO DOS_command
```

If you want to issue a FOR command from the DOS prompt, the only difference is that, rather than preceding the variable name with two percent signs (%%F), you use only one (%F). Therefore, the following DOS command works the same as the previous batch file:

```
A> FOR %F IN (JAN.EXP FEB.EXP MAR.EXP) DO TYPE %F
```

Using Multiple Batch Parameters with SHIFT

In Chapter 11 you learned that DOS supports batch parameters %0 to %9. At that time you found that if you invoked the batch file

```
ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9
```

with more than nine batch parameters, as in

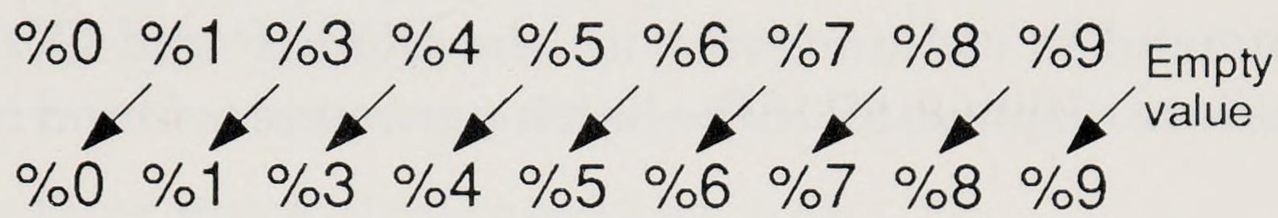
```
A> B:ECHOTEST A B C D E F G H I J K L M
```

the batch file would display only the first nine parameters, as shown here:

```
A> B:ECHOTEST A B C D E F G H I J K L M
A B C D E F G H I

A>
```

Tip: *The DOS SHIFT command allows your batch files to support more than nine batch parameters.*

Each time DOS encounters a SHIFT command within a batch file, it *shifts* (rotates) each parameter one position to the left. In other words, %2 becomes %1, %1 becomes %0, and so on.



Create the batch file

```
ECHO OFF
ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9
SHIFT
ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Next, invoke the procedure with

```
A> B:SHIFTST A B C D E
```

In this case, DOS will display

```
A> B:SHIFTST A B C D E

A> ECHO OFF
A B C D E
B C D E

A>
```

As you can see, following the SHIFT command, DOS has rotated the batch parameters to the left as discussed. If you invoke this procedure with more than nine parameters, as in

```
A> B:SHIFTST A B C D E F G H I J K L M

A> ECHO OFF
A B C D E F G H I
B C D E F G H I J

A>
```

DOS shifts the tenth parameter into %9 following the SHIFT command. Using the SHIFT command in conjunction with the DOS IF and GOTO commands, this batch procedure displays all of the batch parameters found in the command line:

```
ECHO OFF
:REPEAT
ECHO %1
SHIFT
IF NOT '%1'==" GOTO REPEAT
```

If you invoke the procedure with

```
A> B:LIST A B C D E
```

the batch file will display

```
A> B:LIST A B C D E

A> ECHO OFF
A
B
C
D
E

A>
```

To better understand the processing involved, you might want to include the line

```
ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9
```

as shown here:

```
ECHO OFF
:REPEAT
ECHO %1
SHIFT
ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9
IF NOT '%1'==" GOTO REPEAT
```

When you later invoke the procedure with

```
A> B:LIST 1 2 3 4 5
```

the batch file will display

```
A> B:LIST 1 2 3 4 5
1
2 3 4 5
2
3 4 5
3
4 5
4
5
5

A>
```

This output makes the function of the SHIFT command easier to understand.

Remember that the batch file

```
FOR %%F IN (%1) DO TYPE %%F
```

essentially allows you to issue the DOS TYPE command with the DOS wildcard characters.

```
A> T *.*
```

By using the SHIFT command as shown previously, you can enhance this procedure, supporting such commands as

```
A> T CONFIG.SYS *.EXP JULY.*
```

In this case, the batch file will first display the contents of the CONFIG.SYS file, followed by all files with the extension EXP, followed by files with the name JULY:

```
ECHO OFF
:REPEAT
IF '%1'==" GOTO DONE
FOR %%I IN (%1) DO TYPE %%I
SHIFT
GOTO REPEAT
:DONE
```

The processing involved in this batch file looks complex at first. Keep in mind, however, that the file uses only the commands that you have worked with so far in this book.

Named Parameters

In Chapter 10, which discussed the DOS environment, you found that DOS supports the notion of named parameters. Consider this batch file:

```
TYPE %FILE%
```

The characters "%FILE%" compose a named batch parameter. When DOS encounters a named parameter in a batch file, it searches the DOS environment for a matching entry. In this case, DOS will search for an entry in the form

```
FILE=
```

Tip: *DOS named parameters appear in your batch files surrounded by percent signs (%named_parameter%). When DOS encounters a named parameter, it searches the DOS environment for a matching entry.*

If DOS encounters a matching entry in the DOS environment, it substitutes the value associated with the named parameter in the batch command line. Assuming, for example, that the DOS environment contains

```
A> SET
COMSPEC=A:\COMMAND.COM
PATH=
FILE=CONFIG.SYS

A>
```

The previous batch file directs DOS to substitute the named parameter FILE with CONFIG.SYS, resulting in the command

```
TYPE CONFIG.SYS
```

If no matching entry is found in the DOS environment, DOS will issue the command

```
TYPE
```

which results in the error message

```
Invalid number of parameters
```

The following batch file modifies the previous one to obtain the desired screen background color from the DOS environment (as opposed to the batch command line).

```
A> EDLIN  B:NEWCOLOR.BAT
New file
*I
        1:* ECHO OFF
        2:* IF '%COLOR%'==" GOTO DONE
        3:* IF %COLOR%==RED ECHO ^V[[41m
        4:* IF %COLOR%==GREEN ECHO ^V[[42m
        5:* IF %COLOR%==ORANGE ECHO ^V[[43m
        6:* IF %COLOR%==BLUE ECHO ^V[[44m
        7:* :DONE
        8:* ^C
*E

A>
```

In this case, DOS will search the environment for an entry in the form COLOR=. Notice the test within the batch file for a nonexistent entry:

```
IF '%COLOR%'==" GOTO DONE
```

If DOS does not find an entry corresponding to COLOR in the environment, the IF command evaluates as true and the batch file terminates. If you are using a monochrome monitor, use SET to set the COLOR= entry to 1 and DOS will display your text in bold print.

Tip: *The DOS FOR command supports named parameter processing.*

So far, each of the examples that we have used with the DOS FOR command have dealt with file manipulation. For the next example, try something new. Using the DOS SET command, issue the following:

```
A> SET  COLORS=BLUE,GREEN,RED
```

Next, create a batch file called SHOWCLR.BAT, as shown here:

```
A> COPY  CON  B:SHOWCLR.BAT
ECHO OFF
FOR %%I IN (%COLOR%) DO ECHO %%I
^Z
            1 File(s) copied
```

When you invoke this batch file with

```
A> B:SHOWCLR
```

DOS will display

```
A> B:SHOWCLR

A> ECHO  OFF
BLUE
GREEN
RED

A>
```

Note that FOR separates items in the environment with commas, semicolons, or a space. Fixed-disk users with subdirectories can list a directory of the files that DOS will examine in the command path with the batch file

```
FOR %%I IN (%PATH%) DO DIR %%I
```

Invoking Nested Batch Procedures

In some cases, you will want to invoke one batch file from within another batch file. The process of placing a reference to one batch file in a second is called *nesting a batch file.* Unless the reference to the second batch file is the last command in your batch file, as in

```
DATE
TIME
BATFILE
```

you must take special steps before DOS can successfully perform the desired batch processing.

Trap: *Unless your nested batch-file reference is the last command in your batch file, you must use either COMMAND /C or CALL (DOS version 3.3) to invoke the procedure.*

Create the following batch files:

```
A> COPY  CON  B:FIRST.BAT
ECHO OFF
ECHO IN FIRST
B:SECOND
ECHO BACK IN FIRST
^Z
            1 File(s) copied

A> COPY  CON  B:SECOND.BAT
ECHO IN SECOND
^Z
            1 File(s) copied

A>
```

As you can see, the batch file SECOND.BAT is nested within the FIRST.BAT file. When you invoke the FIRST file, as in

```
A> B:FIRST
```

DOS will display

```
A> FIRST

A> ECHO  OFF
IN FIRST
IN SECOND

A>
```

As you can see, DOS never executed the command

```
ECHO BACK IN FIRST
```

You must use either the COMMAND /C (if you are using DOS 3.2 or an earlier version) or the CALL command (for DOS 3.3), as shown here:

```
ECHO OFF
REM DOS 3.2 and Before
ECHO IN FIRST
COMMAND /C B:SECOND
ECHO BACK IN FIRST
```

or

```
ECHO OFF
REM DOS 3.3
ECHO IN FIRST
CALL B:SECOND
ECHO BACK IN FIRST
```

Remember that COMMAND is an external DOS command residing on disk. Once you correctly invoke the nested procedure as just shown, DOS will display

```
A> FIRST

A> ECHO  OFF
IN FIRST
IN SECOND
BACK IN FIRST

A>
```

As you can see, the procedures now work as desired.

AUTOEXEC.BAT: A Special Batch File

Each time DOS starts, it searches your boot disk for the AUTOEXEC.BAT file. If the file does not exist on disk, DOS displays the familar DATE and TIME prompts. However, if DOS finds AUTOEXEC.BAT, it executes the commands that the batch file contains in place of the DATE and TIME commands. The contents of AUTOEXEC.BAT are no different from the other batch files that you have examined. The only way in which AUTOEXEC.BAT differs is that DOS AUTOmatically EXECutes (AUTOEXEC) this batch file each time its starts.

Tip: *Each time DOS starts, it searches the boot disk for the AUTOEXEC.BAT file. If DOS locates this file, it executes the commands that the file contains. If AUTOEXEC.BAT does not exist, DOS executes the familar DATE and TIME commands.*

Since DOS always executes the contents of AUTOEXEC.BAT each time your system starts, it will execute the commands that you place in this file each time it starts. This file, for example, directs DOS to clear the screen display and install a print queue, as discussed earlier in this book:

```
CLS
PRINT /D:LPT1 /Q:32 /B:4096 /M:64
```

Later in this book the DOS PATH and PROMPT commands that most users include in AUTOEXEC.BAT will be discussed.

Tips and Traps Summary

Using IF to Make Decisions

Tip: The IF EXIST FILE condition directs DOS to execute a command only if the specified file resides on disk.

Tip: The *string1==string2* condition directs DOS to perform the specified command only when the characters in the first sequence match the characters in the second sequence exactly.

Trap: For the IF condition *string1==string2* to evaluate as true, the characters in both strings must match in terms of case.

Tip: The IF ERRORLEVEL condition instructs DOS to test the success level of the previous command. As long as the program terminates with a value that is greater than or equal to the value you specify, DOS will execute the given command.

Using the DOS NOT Operator

Tip: The DOS NOT operator instructs DOS to perform the specified command in an IF command when a condition tested fails.

Branching Within Your Batch Files Using GOTO

Tip: The GOTO command directs DOS to skip one or more commands or to repeat a series of DOS commands.

Tip: DOS batch labels appear in a batch file preceded by a colon. When DOS encounters the label, it ignores the line, continuing with the next command in the batch file.

Tips and Traps Summary (*continued*)

Trap: If DOS cannot find the label specified in a GOTO command within the batch file, it will display the error message

```
Label not found
```

and terminate the batch-file processing immediately.

Trap: DOS distinguishes between only the first eight characters of your label names.

Testing for a Nonexistent Parameter

Tip: To test for a nonexistent batch parameter, use the condition IF '%1'==". If you do not specify a batch parameter, the condition becomes IF "==".

Applying a DOS Command to a Collection of Files

Tip: The FOR command instructs DOS to repeat a specific command for a given set of DOS files.

Tip: The FOR command supports DOS wildcard characters.

Tip: Although FOR is most commonly used in DOS batch files, DOS also allows you to invoke FOR from the DOS prompt.

Using Multiple Batch Parameters with SHIFT

Tip: The DOS SHIFT command allows your batch files to support more than nine batch parameters.

Tips and Traps Summary (*continued*)

Named Parameters

Tip: DOS named parameters appear in your batch files surrounded by percent signs (%named_parameter%). When DOS encounters a named parameter, it searches the DOS environment for a matching entry.

Tip: The DOS FOR command supports named parameter processing.

Invoking Nested Batch Procedures

Trap: Unless your nested batch-file reference is the last command in your batch file, you must use either COMMAND /C or CALL (DOS version 3.3) to invoke the procedure.

Tip: Each time DOS starts, it searches the boot disk for the AUTOEXEC.BAT file. If DOS locates this file, it executes

13

Using DOS Subdirectories

So far, all of the example commands have been floppy-disk based for two reasons. First, not all users have a fixed disk. Second, since file management has not yet been discussed, you could have cluttered up your fixed disk with temporary example files.

In this chapter you will learn how to increase your file and disk organization with DOS subdirectories. In addition, you will learn how to increase the number of files that your disks can store.

Getting Started with DOS Subdirectories

Trap: *DOS disks, regardless of the amount of free space they contain, are restricted to a set number of files.*

As you have seen, each time DOS displays a directory listing, the last item of information that DOS displays is the amount of free space that the disk contains.

```
34 File(s)  4096 bytes free
```

Obviously, you cannot place more information on a disk than the disk has available space for. In a similar manner, unless you use DOS subdirectories as discussed in this chapter, your disks are restricted to the number of files shown in Table 13-1.

Therefore, regardless of the amount of free space your disk contains, by default, DOS will not create more files on the disk than shown in this table. If you are using a 360K disk, for example, and you attempt to create a 113th file on the disk, DOS will display

```
File creation error
```

Even fixed-disk users with as much as 30MB of free disk space (space for 30 million characters) are still restricted to a set number of files unless they use DOS subdirectories.

Table 13-1. Subdirectories Supported by Root Directory

Disk Space	Maximum Number of Subdirectories in the Root Directory
160K	64
180K	
320K	112
360K	
1.2MB	224
Fixed disk	Based on partition size

Tip: *DOS subdirectories allow DOS to place an unlimited number of files on a disk. With subdirectories, the only restriction to the number of files that you can create is available disk space.*

The second major advantage that DOS subdirectories provide you is file organization. If you have been using DOS for some time, you are probably well aware of how the number of your DOS files can quickly grow, which makes locating a specific file on your disk difficult at best. Just as you might separate the files in your office into different drawers of a filing cabinet, DOS allows you to group logically related files in different storage locations on your disk. This in turn improves your file organization, while allowing your disk to store a greater number of files.

Each time DOS displays a directory listing of your disk, it displays the list of files that reside on the disk. This list of files is called a *directory*.

```
Volume in drive A is MS 330PP01
Directory of A:\
```

Directory

To organize the files on your disk, you will create DOS subdirectories, which are nothing more than sublists of related files. To create a DOS subdirectory, you use the DOS MKDIR command.

Using MKDIR to Create DOS Subdirectories

Place a newly formatted disk in drive A and issue the DOS DIR command, as shown here:

```
A> DIR

Volume in drive A has no label
Directory of  A:\

File not found
```

In this case, the directory (list of files) is empty, which makes sense because you have not yet created any files on the disk.

Tip: *Each DOS disk has a directory, called the root directory, from which all subdirectories grow.*

Recall from earlier in this chapter that a DOS directory is nothing more than a list of files, and that a subdirectory is simply a sublist of related files. Each DOS disk has one directory (called the root directory) that you can divide into sublists or *subdirectories*. This directory is called the root directory since, pictorially, all of your other directories appear to grow from the root in a tree-like manner. DOS uses the backslash character (\) to represent the root directory. If you examine the previous directory listing, DOS states that it is listing the contents of the root directory.

```
Volume in drive A is MS 330PP01
Directory of  A:\
```

Root directory drive A

Tip: *The MKDIR command instructs DOS to create a subdirectory with the name specified in the command line. MKDIR is an internal DOS command.*

For the following examples, assume that you want to create three DOS subdirectories to logically group files containing your monthly bills, class notes from school, and a catchall directory for miscellaneous files. Your subdirectories will be named as follows:

BILLS	Contains your monthly bills
SCHOOL	Contains your school class notes
MISC	Contains miscellaneous files

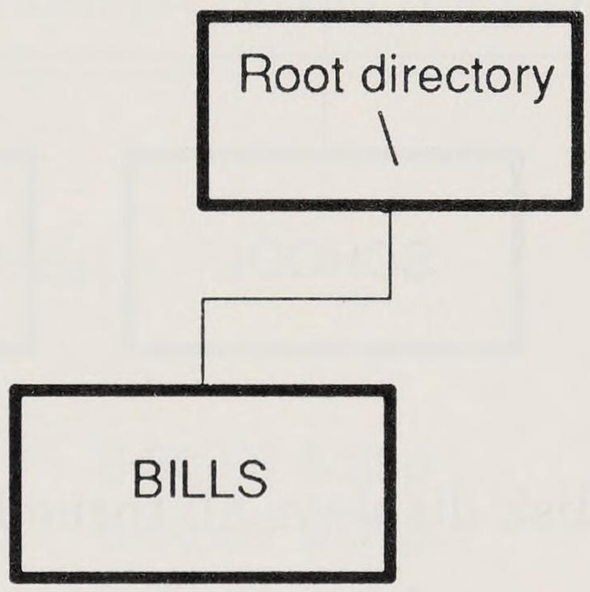
To create a DOS subdirectory, place the directory name in the MKDIR command line, as shown here.

```
A> MKDIR \BILLS
A>
```

Notice the backslash that precedes the directory name BILLS. As just discussed, DOS defines the backslash (\) as the root directory. The previous command tells DOS to create a subdirectory called BILLS in the root directory. Later in this chapter you will learn that in some cases you will not have to specify the backslash in front of your directory names while in other cases you will. For simplicity for now, include the backslash in your subdirectory commands.

A directory listing of the disk in drive A now reveals the newly created directory. Notice that DOS no longer displays the number of bytes in the directory file, but instead displays <DIR>, which tells you that the entry is a DOS subdirectory as opposed to a standard DOS file.

Pictorially, the directories on the disk now appear as



Repeat this process creating a directory called SCHOOL.

```
A> MKDIR \SCHOOL
```

A directory listing of the disk now reveals

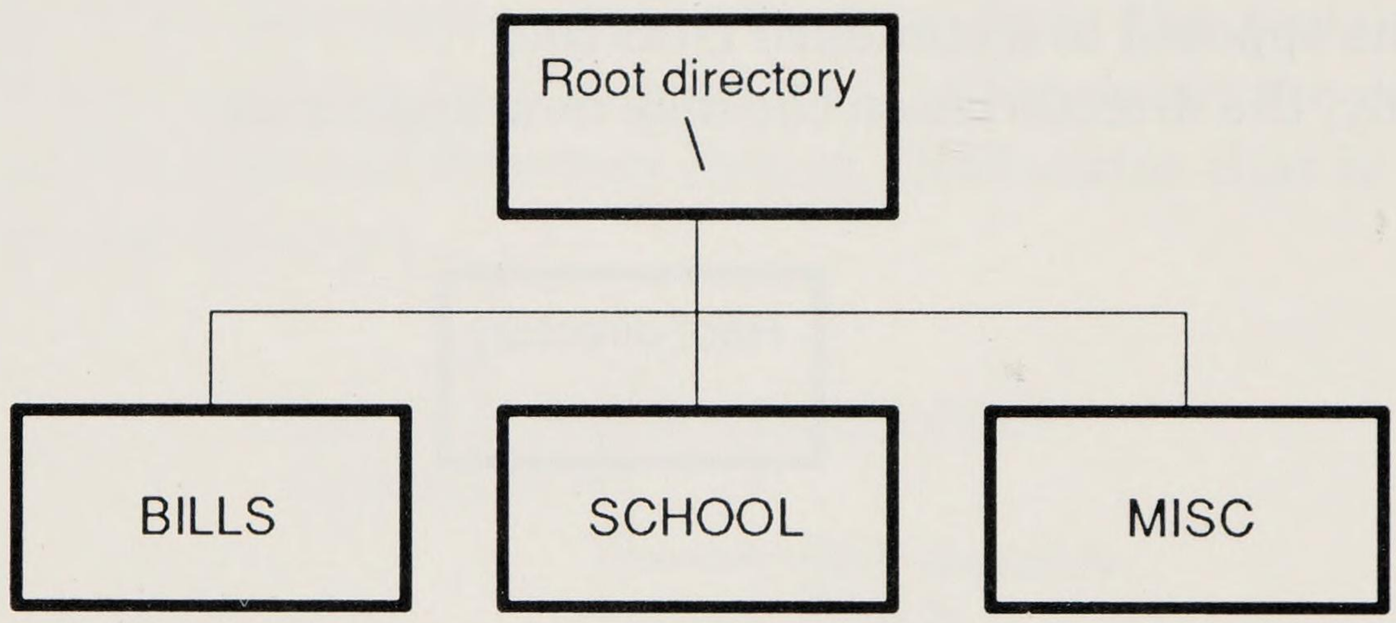
```
A> DIR

Volume in drive A has no label
Directory of  A:\

BILLS      <DIR>                6-30-88  5:39p
SCHOOL     <DIR>                6-30-88  5:40p
           2 File(s)      361472 bytes free
```

Tip: *Due to its frequent use, DOS allows you to abbreviate the MKDIR command to MD.*

Using the MD abbreviation, create your final subdirectory (MISC), as shown here:

```
A> MD \MISC
```

Pictorially, the directory structure now appears as



Listing the files on the disk displays all three DOS subdirectories.

```
A> DIR

Volume in drive A has no label
Directory of  A:\

BILLS      <DIR>                 6-30-88  5:39p
SCHOOL     <DIR>                 6-30-88  5:39p
MISC       <DIR>                 6-30-88  5:39p
           3 File(s)     359424 bytes free
```

Tip: *DOS directory names follow the same rules as DOS file names—an 8-character name followed by an optional 3-character extension.*

Most users don't realize that they can use the 3-character extension in their DOS directory names. Since DOS allows you to use up to 11-character names, you can create meaningful subdirectory names, as shown here:

INCOME-88.TAX	BUSINESS.EXP	TRAVEL.CST
STOCKS.PFT	MEETINGS.NTS	BUILDING.NTS

Trap: *If MKDIR cannot create the specified directory, it will display the message*

```
Unable to create directory
```

This error message is most common when the directory that you specify already exists on disk. For example, given the DOS subdirectories

```
A> DIR

Volume in drive A has no label
Directory of  A:\

BILLS       <DIR>          6-30-88  5:39p
SCHOOL      <DIR>          6-30-88  5:39p
MISC        <DIR>          6-30-88  5:39p
            3 File(s)    359424 bytes free
```

the command

```
A> MKDIR \BILLS
```

fails, and DOS will display

```
Unable to create directory
```

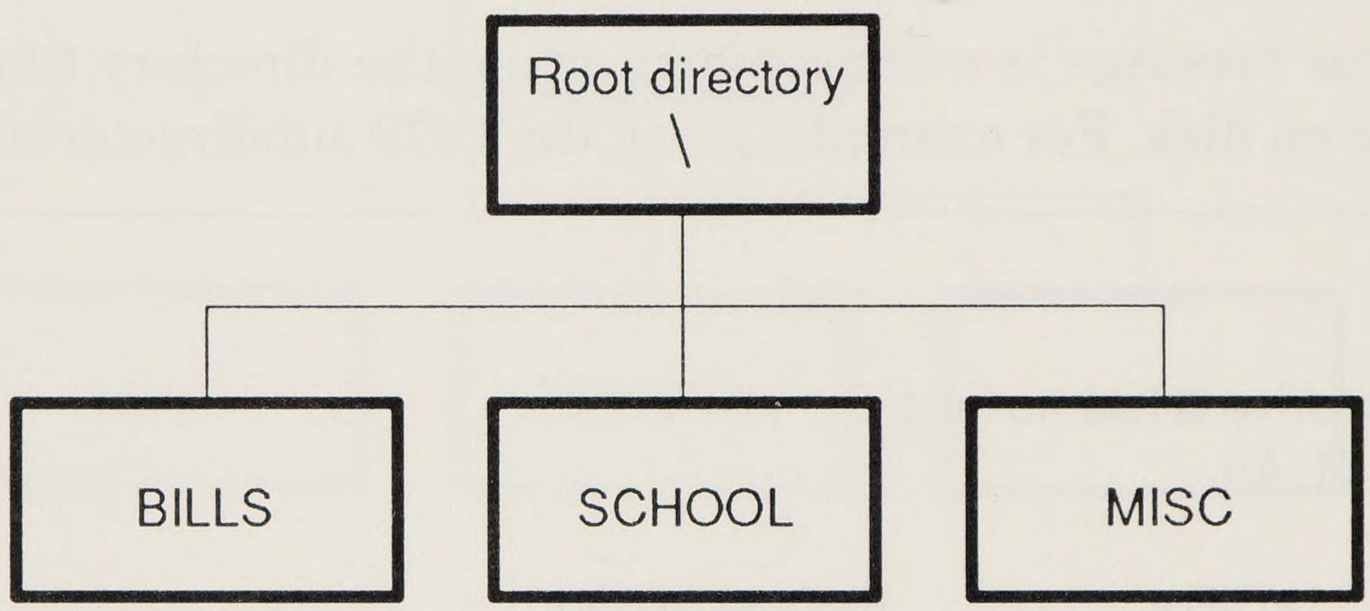
Just as DOS defines the current disk drive as the disk that, by default, DOS will examine for external commands, DOS also defines the current directory. Throughout this book, the root is the only directory that has been used for

the current directory. The DOS CHDIR command allows you to set the current directory.

Selecting the Current Directory with CHDIR

Tip: *The DOS CHDIR command instructs DOS to select the directory specified in the command line as the current directory.*

Given the previous directory structure



the command

```
A> CHDIR \BILLS
A>
```

directs DOS to select the BILLS subdirectory as the current directory. A directory listing of drive A reveals

```
A> DIR

Volume in drive A has no label
Directory of  A:\BILLS

.          <DIR>          6-30-88  5:39p
..         <DIR>          6-30-88  5:39p
           2 File(s)    359424 bytes free
```

Note that DOS tells you it is now examining the BILLS subdirectory.

```
Volume in drive A has no label
Directory of  A:\BILLS
```

Later, this chapter will examine the two directories:

```
.          <DIR>        6-30-88  5:39p
..         <DIR>        6-30-88  5:39p
```

For now, you need to know only that DOS places these two subdirectories in each DOS directory you create. Create the following file.

```
A> COPY  CON  BILLS.NTS
The subdirectory BILLS contains
my monthly expenses.
^Z
            1 File(s) copied

A>
```

A directory listing of BILLS now reveals

```
A>

Volume in drive A has no label
Directory of  A:\BILLS

.          <DIR>                 6-30-88  5:39p
..         <DIR>                 6-30-88  5:39p
BILLS      NTS          55       6-30-88  5:44p
           3 File(s)    358400 bytes free
```

Using CHDIR, select the root directory as your current directory.

```
A> CHDIR \
```

As you can see, selecting the root directory is no different from selecting any DOS subdirectory. A directory list of your disk now reveals

```
A>

Volume in drive A has no label
Directory of  A:\

BILLS      <DIR>              6-30-88  5:39p
SCHOOL     <DIR>              6-30-88  5:39p
MISC       <DIR>              6-30-88  5:39p
           3 File(s)     358400 bytes free
```

Notice that the BILLS.NTS file does not appear in the directory listing.

Trap: *Unless told otherwise, DIR displays only the files contained in the current directory.*

Later in this chapter you will learn how to perform directory listings of any DOS subdirectory.

Tip: *Because of its frequent use, DOS allows you to abbreviate CHDIR to CD.*

Using this form of the CHDIR command, select the SCHOOL directory as the current directory.

```
A> CD \SCHOOL
A>
```

A directory listing of your disk reveals

```
A> DIR

Volume in drive A has no label
Directory of  A:\SCHOOL

.          <DIR>              6-30-88  5:39p
..         <DIR>              6-30-88  5:39p
           2 File(s)     358400 bytes free
```

As you can see, this directory contains the subdirectories . and .. as just discussed. Next, select the \MISC subdirectory as the current directory.

```
A> CD \MISC
A>
```

As you can see, DOS allows you to select a different directory as the current directory, regardless of your current directory. Using the DOS COPY command, create a file called NOTES.TXT.

```
A> COPY CON NOTES.TXT
The subdirectory MISC contains
miscellaneous data files.
^Z
            1 File(s) copied
```

Listing the files in your directory results in

```
A> DIR

Volume in drive A has no label
Directory of A:\MISC

NOTES     TXT     61          6-30-88 5:47p
          3 File(s)     357376 bytes free
```

Trap: *If the DOS subdirectory specified in the command line does not exist, CHDIR will display the message*

```
Invalid directory
```

If this message occurs, make sure that you have not mistyped the directory name, or forgotten to precede the directory name with a backslash.

Tip: *If you enter the CHDIR command without a directory name in the command line, CHDIR will display the current directory.*

At the DOS prompt, type **CHDIR** and press ENTER.

```
A> CHDIR
```

CHDIR displays the current directory.

```
A> CHDIR
A:\MISC

A>
```

Tip: *DOS maintains a current directory for each disk drive.*

Since your computer has multiple disk drives, DOS keeps track of the subdirectory that you are currently using on each disk. Therefore, you can set or display the current directory for a disk drive other than the current default simply by preceding the directory name with a disk drive specifier, as shown here:

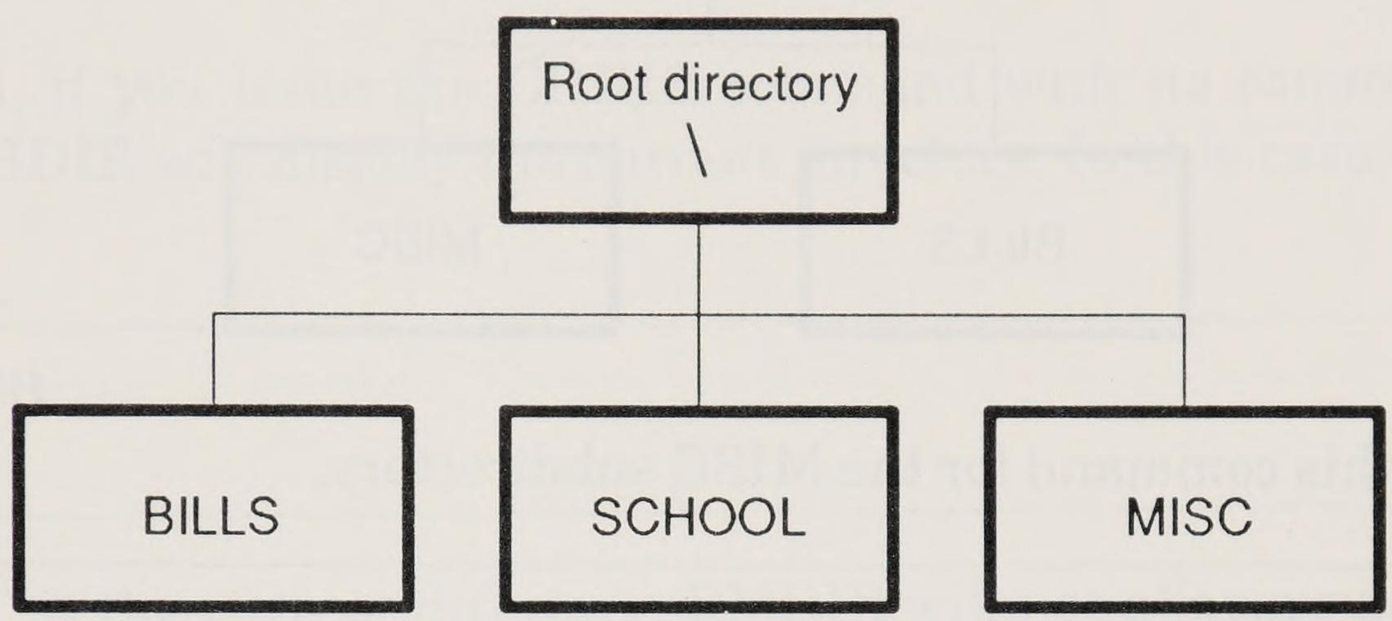
```
A> CHDIR  B:EXAMPLES
A> CHDIR  B:
B:\EXAMPLES

A>
```

Removing Subdirectories With RMDIR

As stated, DOS treats subdirectories differently than files. You cannot use the DOS DEL command to remove a DOS subdirectory from your disk. Instead, you must use the RMDIR command.

Tip: *The DOS RMDIR command instructs DOS to remove the directory specified in the command line from your disk.*

Pictorially, your current directory structure contains



Select the root directory as your current directory.

```
A> CHDIR \
```

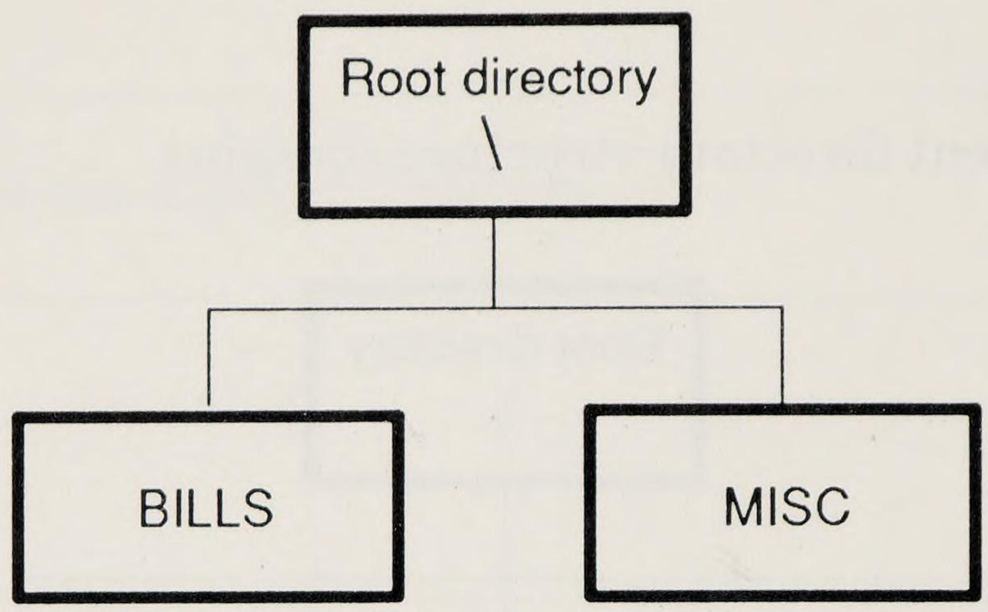
Next, use the DOS RMDIR command to remove the SCHOOL subdirectory.

```
A> RMDIR \SCHOOL
A>
```

Listing the files on your disk reveals

```
A> DIR

Volume in drive A has no label
Directory of  A:\

BILLS        <DIR>          6-30-88  5:39p
MISC         <DIR>          6-30-88  5:39p
             2 File(s) 358400 bytes free
```

Pictorially, the directory structure has become



Repeat this command for the MISC subdirectory.

```
A> RMDIR \MISC
```

In this case, the DOS RMDIR command fails and DOS displays the message

```
Invalid path, not directory,
or directory not empty
```

Trap: *RMDIR will not remove a subdirectory that contains files.*

To prevent you from inadvertently deleting needed files, RMDIR will not remove a subdirectory that contains files. Therefore, select the MISC subdirectory as the current directory and delete the files that it contains.

```
A> CHDIR \MISC
A> DEL *.*
```

Again, issue the RMDIR command

```
A> RMDIR \MISC
```

Again the command fails and DOS displays

```
Invalid path, not directory,
or directory not empty
```

Trap: *DOS does not allow you to delete the current directory.*

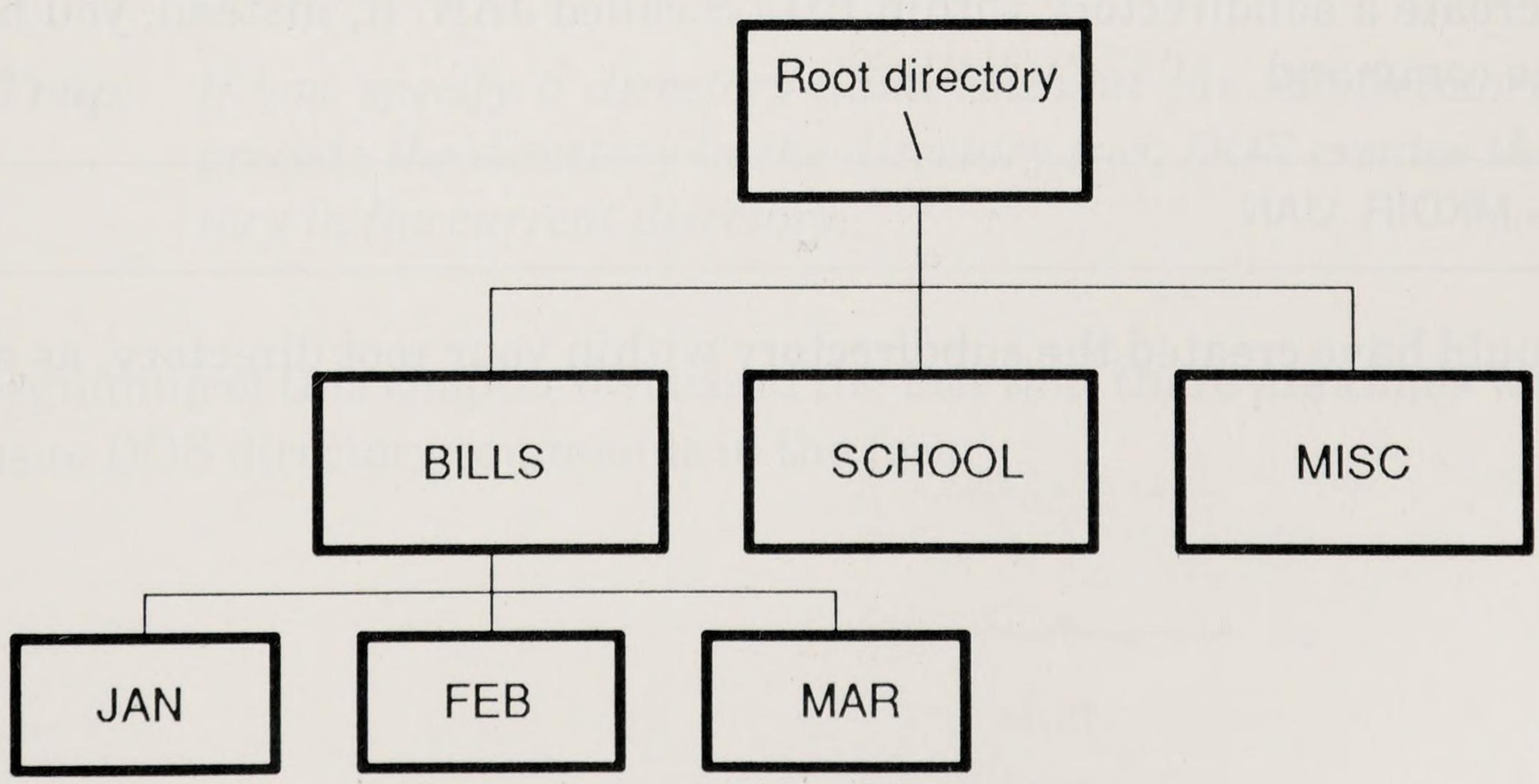
As discussed, if you issue the CHDIR command with no command-line parameters, CHDIR will display the current directory. In this case, CHDIR will display

```
A> CHDIR
A:\MISC
```

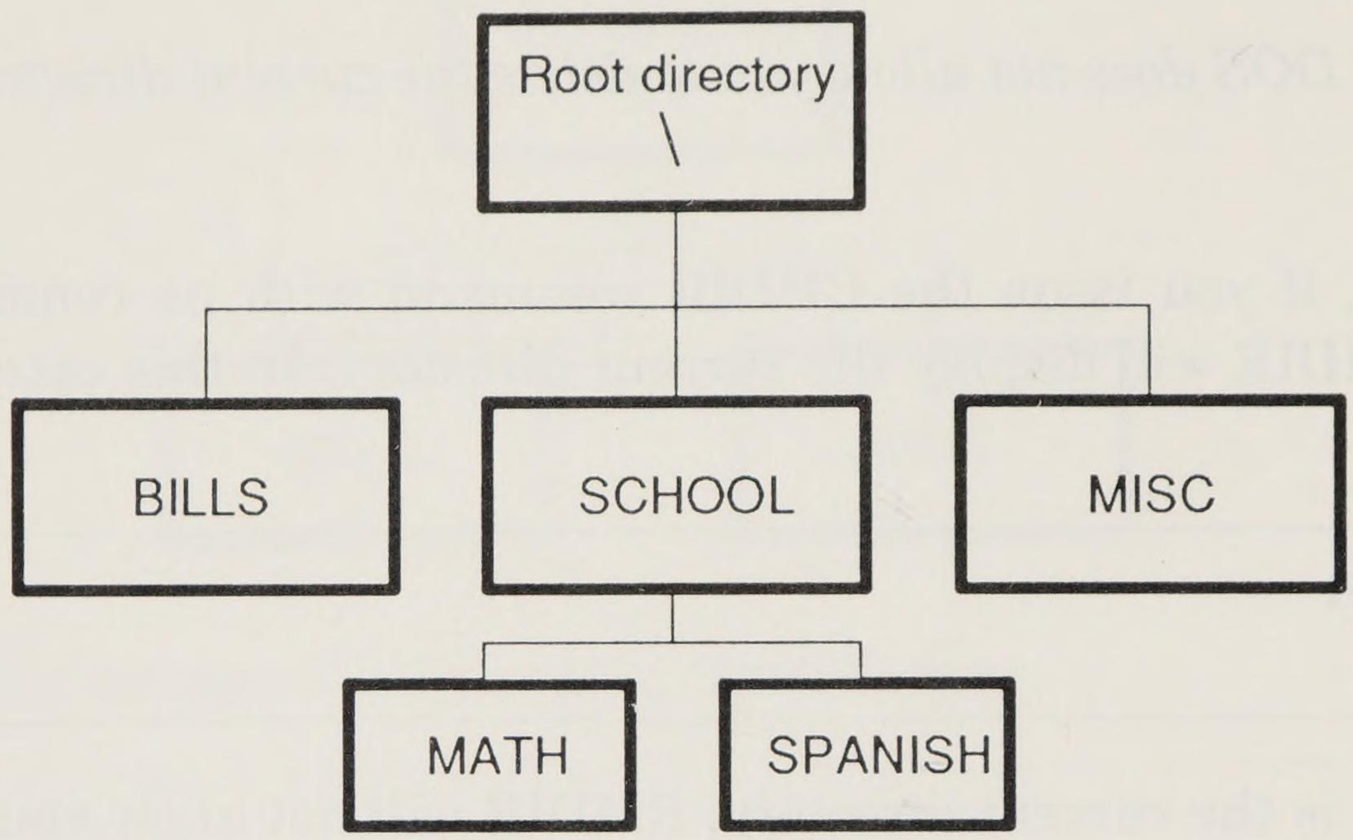
Since \MISC is the current directory, RMDIR will not allow you to remove it. Instead, you must first select a different directory as the current directory before you can issue this command.

Creating Multiple Levels of Subdirectories

Just as DOS allows you to create subdirectories within the root directory, it also allows you to logically divide all of your subdirectories into additional directories to further improve your disk organization. Assume, for example, that you want to track your bills by month. Consider the directory structure



Likewise, to divide the SCHOOL subdirectory into specific class subdirectories, your structure might become



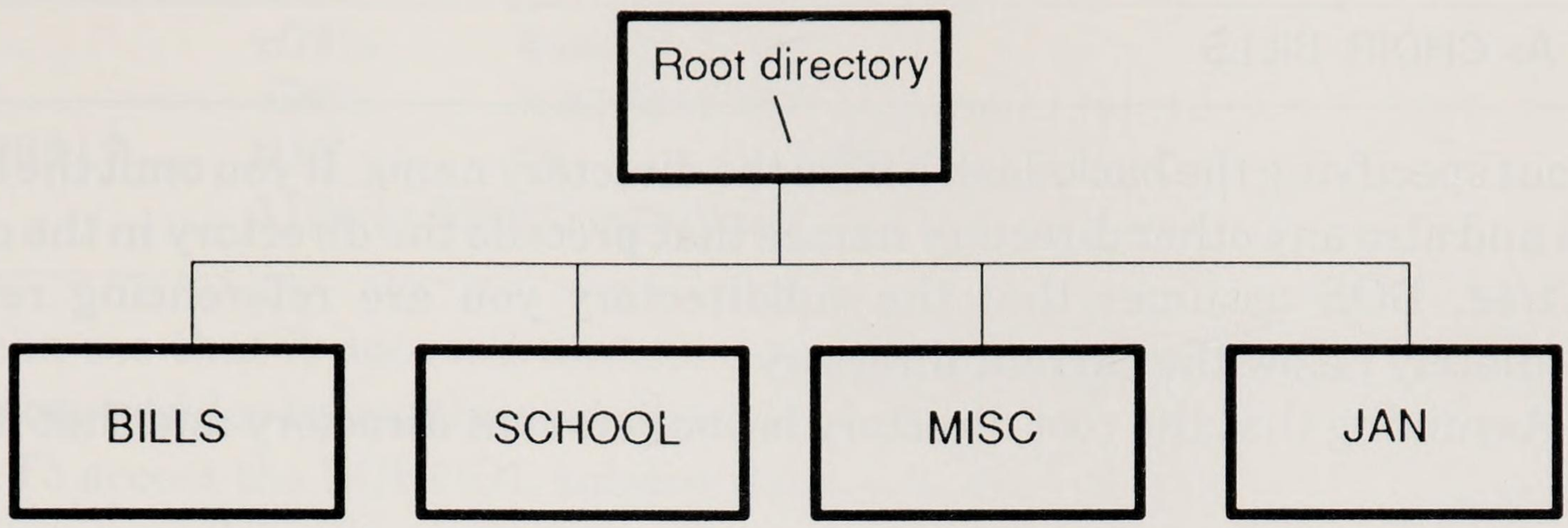
To create these DOS subdirectories, you use the DOS MKDIR command. In this case, your commands would take the form

```
A> MKDIR \BILLS\JAN
A> MKDIR \BILLS\FEB
A> MKDIR \BILLS\MAR
A> MKDIR \SCHOOL\MATH
A> MKDIR \SCHOOL\SPANISH
```

The directory name \BILLS\JAN, for example, directs DOS to start in the root directory (\), to traverse the BILLS subdirectory (assuming that it exists), and to create a subdirectory within BILLS called JAN. If, instead, you had issued the command

```
A> MKDIR \JAN
```

DOS would have created the subdirectory within your root directory, as shown here.



Trap: *If a directory in the series of directories given does not exist, DOS will display the message*

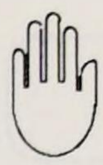
```
Unable to create directory
```

Given the command

```
A> MKDIR \BILLS\JAN
```

before DOS can create the subdirectory JAN, the BILLS subdirectory must already exist on disk. To select this subdirectory, use the command

```
A> CHDIR \BILLS\JAN
```



Trap: *If you specify a directory name without the subdirectories that precede the directory in the directory tree, DOS creates the directory in the current directory.*

The beginning of this chapter discussed the fact that there are times when you can issue DOS directory commands in the form

```
A> CHDIR BILLS
```

without specifying the backslash before the directory name. If you omit the backslash and also any other directory names that precede the directory in the directory tree, DOS assumes that the subdirectory you are referencing resides immediately below the current directory.

Assuming that the root directory is your current directory and that it contains

```
A> DIR

Volume in drive A has no label
Directory of  A:\

BILLS      <DIR>        6-30-88  5:39p
SCHOOL     <DIR>        6-30-88  5:56p
MISC       <DIR>        6-30-88  5:39p
           3 File(s)    358400 bytes free
```

the command

```
A> CHDIR BILLS
```

succeeds, because BILLS resides immediately below the current directory. However, once you select BILLS as the current directory, the command

```
A> CHDIR SCHOOL
```

fails and DOS displays the message

```
Invalid directory
```

If you issue a directory listing of the files in the BILLS subdirectory

```
A> DIR

Volume in drive A has no label
Directory of  A:\BILLS
```

```
.          <DIR>      6-30-88  5:39p
..         <DIR>      6-30-88  5:39p
BILLS      NTS        55          6-30-88  5:44p
           3 File(s) 358400 bytes free
```

you can see that it does not contain a subdirectory called SCHOOL (that subdirectory resides below the root directory).

To access the SCHOOL subdirectory, you have two choices. First, you can specify a complete directory name (including the backslash).

```
A> CHDIR \SCHOOL
```

Second, you can select the root directory as the current directory

```
A> CHDIR \
```

and then again issue the command

```
A> CHDIR SCHOOL
```

In this case, the command will succeed, since the SCHOOL subdirectory resides below the root directory.

Trap: *Although DOS allows you to create a virtually unlimited number of subdirectories, the longest directory string that DOS will process is 63 characters.*

As you just saw, DOS allows you to logically divide your directories to increase your file organization. The restriction DOS places upon the number of subdirectories you can use is that the longest directory string it will process is 63 characters.

```
A>CD\SCHOOL\SPANISH\OCTOBER\NOTES\MAYAN\EMPIRE\CULTURE\LANGUAGE.TXT
```

Rules for Naming Your Directories

As you create your DOS directories, keep the following rules in mind:

- DOS subdirectories follow the same naming conventions as DOS files. The following characters are valid in both DOS file and subdirectory names:

  ~!@#$%^&*()-_{}'

- If you don't start the directory name at the root directory by placing a backslash (\) at the start of the name, DOS will create the subdirectory within the current directory.

- By including a disk drive specifier, DOS allows you to create a directory on a disk other than the current default disk:

```
A> MKDIR B:\TEST
```

- Do not create a DOS directory with the same name as a file that will reside in the same directory.

- Do not create a subdirectory called \DEV. DOS uses a hidden internal directory called \DEV to perform input and output operations to devices (such as your printer).

- Although DOS allows you to create a virtually unlimited number of subdirectories, the longest directory name DOS supports is 63 characters.

- If you don't use DOS subdirectories, the maximum number of files that you can place on your disk is fixed by DOS.

What Are . and ..?

As stated, every DOS subdirectory contains the two directories shown here:

```
.        <DIR>        6-30-88  5:39p
..       <DIR>        6-30-88  5:39p
```

To help you issue directory-manipulation commands, DOS predefines the period and double period as follows:

.	An abbreviation for the current directory
..	An abbreviation for the directory immediately above the current directory

Given the current directory \BILLS, DOS defines the period and double period as follows:

.	The subdirectory \BILLS
..	The root directory, since it is immediately above BILLS

Likewise, if the directory \BILLS\JAN is the current directory, DOS will define the period and double period as follows:

.	The subdirectory \BILLS\JAN
..	The subdirectory \BILLS, since it is the directory immediately above the current directory

Verify this by selecting the BILLS directory as the current directory.

```
A> CHDIR \BILLS
```

A directory listing of this subdirectory displays

```
A> DIR

Volume in drive A has no label
Directory of  A:\BILLS

BILLS       NTS          55   6-30-88  5:44p
            1 File(s)    358400 bytes free
```

Using the period, display the files in the current directory.

```
A> DIR .

Volume in drive A has no label
Directory of  A:\BILLS

BILLS       NTS          55  6-30-88  5:44p
            1 File(s) 358400 bytes free
```

In a similar manner, use the double period to display the files in the root directory (since the root is immediately above the BILLS subdirectory).

```
A> DIR ..

Volume in drive A has no label
Directory of  A:\

BILLS       <DIR>            6-30-88  5:39p
SCHOOLS     <DIR>            6-30-88  5:56p
MISC        <DIR>            6-30-88  5:39p
            3 File(s) 358400 bytes free
```

Again, using the double period, select the root directory as your current directory.

```
A> CHDIR
A:\BILLS
A> CHDIR ..
A> CHDIR
A:\
A>
```

Most users don't use these two abbreviations often. However, you should know what these abbreviations mean, since they will appear in all of your subdirectories.

Directory Listings of Files Contained in DOS Subdirectories

Earlier, when you listed the contents of the root directory on drive A, you found that unless you told DOS to do otherwise, DOS displayed only files residing in the current directory.

Tip: *To list the files that reside in a DOS subdirectory, place the desired directory name in the DIR command line.*

The DOS DIR command allows you to list the contents of a DOS subdirectory by specifying the name of the subdirectory in the DIR command line, as shown here:

```
A> DIR \BILLS
```

In this case, DOS will respond with

```
Volume in drive A has no label
Directory of  A:\BILLS

.          <DIR>          6-30-88  5:39p
..         <DIR>          6-30-88  5:39p
BILLS      NTS            55  6-30-88  5:44p
           3 File(s) 358400 bytes free
```

Likewise, to display the files contained in the MISC subdirectory, use the command

```
A> DIR \MISC
```

If you are interested in a specific file, use

```
A> DIR \BILLS\BILLS.NTS
```

In this case, DOS will display

```
A> DIR \BILLS\BILLS.NTS

Volume in drive A has no label
Directory of A:\BILLS

BILLS      NTS        55  6-30-88  5:44p
           1 File(s)     358400 bytes free
```

As you can see, to manipulate files contained in DOS subdirectories, you need to tell DOS in which subdirectory the file resides.

Manipulating Files Contained in DOS Subdirectories

Earlier in this book the DOS file-manipulation commands were discussed. COPY, DEL, RENAME, and TYPE are the majority of the DOS commands that you will use on a daily basis. As you will see, these commands are easily applied to files that reside in DOS subdirectories.

To start, create the following file in the MISC subdirectory. (If the MISC directory no longer resides on your disk, create it using MKDIR.)

```
A> COPY CON \MISC\DOS.NTS
DOS subdirectories increase
your file organization.
^Z
          1 File(s) copied
```

If you issue a directory listing of the MISC subdirectory

```
A> DIR \MISC
```

DOS diplays the newly created file. To display the contents of this file, use the DOS TYPE command.

```
A> TYPE \MISC\DOS.NTS
DOS subdirectories increase
your file organization.

A>
```

Use the DOS COPY command to copy this file to the BILLS subdirectory.

```
A> COPY \MISC\DOS.NTS \BILLS\DOS.NTS
```

Again, earlier in this book you found that the DOS COPY command allows you to abbreviate COPY commands when the destination file name is identical to the source file.

```
A> COPY FILENAME.EXT B:
```

This same technique holds true for files that reside in DOS subdirectories. In this case, since the destination file is still to be named DOS.NTS

```
A> COPY \MISC\DOS.NTS \BILLS\DOS.NTS
```

you can simplify the command to

```
A> COPY \MISC\DOS.NTS \BILLS
```

For that matter, the DOS wildcard characters are still fully supported.

```
A> COPY \MISC\DOS.NTS \BILLS\*.*
```

The DOS RENAME command is the only file-manipulation command that sometimes gives users difficulty when they are working with DOS subdirectories. Just as DOS does not allow you to rename a file from one DOS disk to another, DOS does not allow you to rename a file from one DOS subdirectory to another.

Trap: *The DOS RENAME command does not allow you to rename files from one DOS subdirectory to another.*

If you attempt to do so

```
A> RENAME \BILLS\DOS.NTS \MISC\DOS.NTS
```

the command will fail and DOS will display

```
Invalid parameter
```

Likewise, if you even specify the same directory name as the source directory in your target file name

```
A> RENAME \BILLS\DOS.NTS \BILLS\DOS.TXT
```

the command will fail and DOS will again display

```
Invalid parameter
```

The correct command in this case is

```
A> RENAME \BILLS\DOS.NTS DOS.TXT
```

regardless of your current directory. Since DOS will not allow you to rename a file to a different directory, DOS automatically knows in this case that the DOS.TXT file should reside in the BILLS subdirectory.

Entering External Commands That Reside in DOS Subdirectories

Tip: *DOS defines a path as the series of directories that it must traverse to find a file or command on disk.*

If you examine the DOS external commands in the Command Reference section (Appendix B) of this book, you will find that the format of most commands is

```
[drive:][path]command_name [parameters]
```

As you have learned, items contained within the left and right brackets are optional. Therefore, you can enter a command as

```
A> DISKCOPY
```

assuming that the DISKCOPY.COM file resides in the current directory of the default disk.

You enter the command as

```
A> B:DISKCOPY
```

if the DISKCOPY.COM file resides in the current directory of the disk in drive B.

If your external command file resides in a DOS subdirectory, precede the command name with the subdirectory path that defines its location, as shown here:

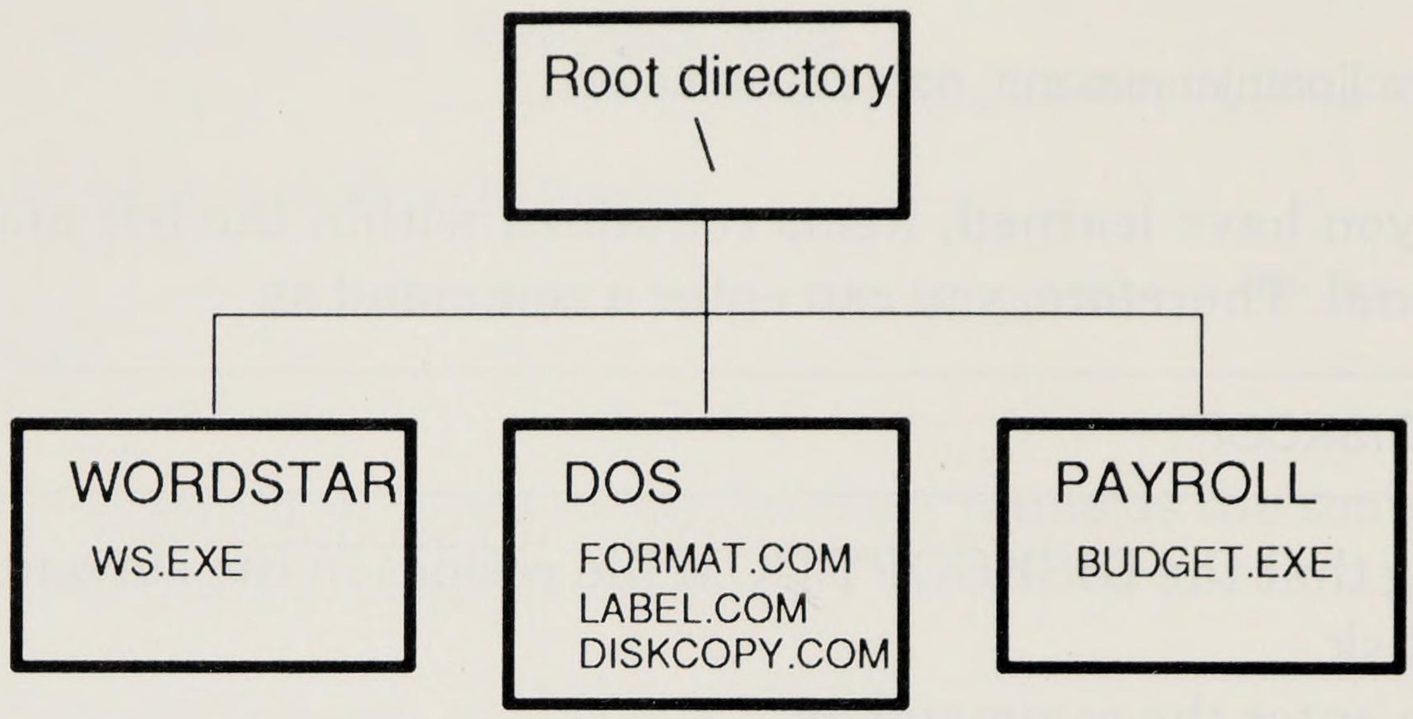
```
C> \DOS\DISKCOPY
```

In this case, the command is assumed to reside in a subdirectory called DOS on drive C.

Tip: *To execute an external DOS command that resides in a DOS subdirectory, precede the command name with a DOS path that defines the command's location on disk.*

In Chapter 14, when installing DOS on your fixed disk is discussed, the ability to execute commands that reside in DOS subdirectories will become paramount.

For now, assuming that your directory structure contains



the following are valid DOS command lines to invoke the programs shown:

```
\WORDSTAR\WS
\DOS\FORMAT
\DOS\LABEL
\DOS\DISKCOPY
\PAYROLL\BUDGET
```

Tips and Traps Summary

Getting Started with DOS Subdirectories

Trap: DOS disks, regardless of the amount of free space they contain, are restricted to a set number of files.

Tip: DOS subdirectories allow DOS to place an unlimited number of files on a disk. With subdirectories, the only restriction to the number of files that you can create is available disk space.

Using MKDIR to Create DOS Subdirectories

Tip: Each DOS disk has a directory, called the root directory, from which all subdirectories grow.

Tip: The MKDIR command instructs DOS to create a subdirectory with the name specified in the command line. MKDIR is an internal DOS command.

Tip: Due to its frequent use, DOS allows you to abbreviate the MKDIR command to MD.

Tip: DOS directory names follow the same rules as DOS file names—an 8-character name followed by an optional 3-character extension.

Trap: If MKDIR cannot create the specified directory, it will display the message

```
Unable to create directory
```

Selecting the Current Directory with CHDIR

Tip: The DOS CHDIR command instructs DOS to select the directory specified in the command line as the current directory.

Tips and Traps Summary (*continued*)

Trap: Unless told otherwise, DIR displays only the files contained in the current directory.

Tip: Because of its frequent use, DOS allows you to abbreviate CHDIR to CD.

Trap: If the DOS subdirectory specified in the command line does not exist, CHDIR will display the message

```
Invalid directory
```

Tip: If you enter the CHDIR command without a directory name in the command line, CHDIR will display the current directory.

Tip: DOS maintains a current directory for each disk drive.

Removing Subdirectories with RMDIR

Tip: The DOS RMDIR command instructs DOS to remove the directory specified in the command line from your disk.

Trap: RMDIR will not remove a subdirectory that contains files.

Trap: DOS does not allow you to delete the current directory.

Creating Multiple Levels of Subdirectories

Trap: If a directory in the series of directories given does not exist, DOS will display the message

```
Unable to create directory
```

Trap: If you specify a directory name without the subdirectories that precede the directory in the directory tree, DOS creates the directory in the current directory.

Tips and Traps Summary (*continued*)

Trap: Although DOS allows you to create a virtually unlimited number of subdirectories, the longest directory string that DOS will process is 63 characters.

Directory Listings of Files Contained in DOS Subdirectories

Tip: To list the files that reside in a DOS subdirectory, place the desired directory name in the DIR command line.

Manipulating Files Contained in DOS Subdirectories

Trap: The DOS RENAME command does not allow you to rename files from one DOS subdirectory to another.

Entering External Commands That Reside in DOS Subdirectories

Tip: DOS defines a path as the series of directories that it must traverse to find a file or command on disk.

Tip: To execute an external DOS command that resides in a DOS subdirectory, precede the command name with a DOS path that defines the command's location on disk.

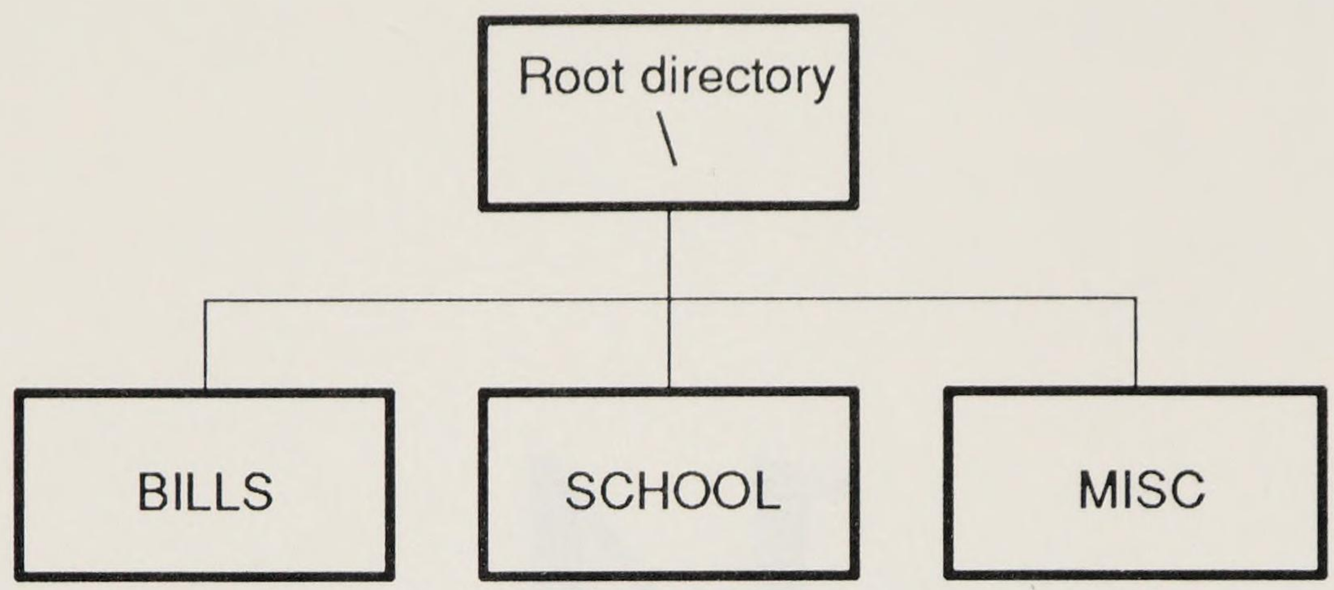
14

Advanced Use of Subdirectories

In Chapter 13 you learned that DOS allows you to organize your files by placing related files into DOS subdirectories. In this chapter you will build upon your knowledge of DOS subdirectories as you examine several additional DOS subdirectory commands and techniques. If you are using a fixed disk, or would like to install DOS on your fixed disk, this chapter will ensure that your fixed disk is optimally organized.

Taking a Look at Your Directory Structure

In Chapter 13 you created a simple directory structure containing



In this case, you know that your disk contains only these three subdirectories, because you just created them. However, if you have been away from this disk for a while, and you need to once again understand the disk's structure, one method is to examine each subdirectory one at a time with the DOS DIR command.

```
A> DIR \MISC
```

In so doing, you can eventually display all of the files that reside on the disk. However, if your directory structure is more complex, as shown in Figure 14-1, using the DOS DIR command in this manner could take considerable time.

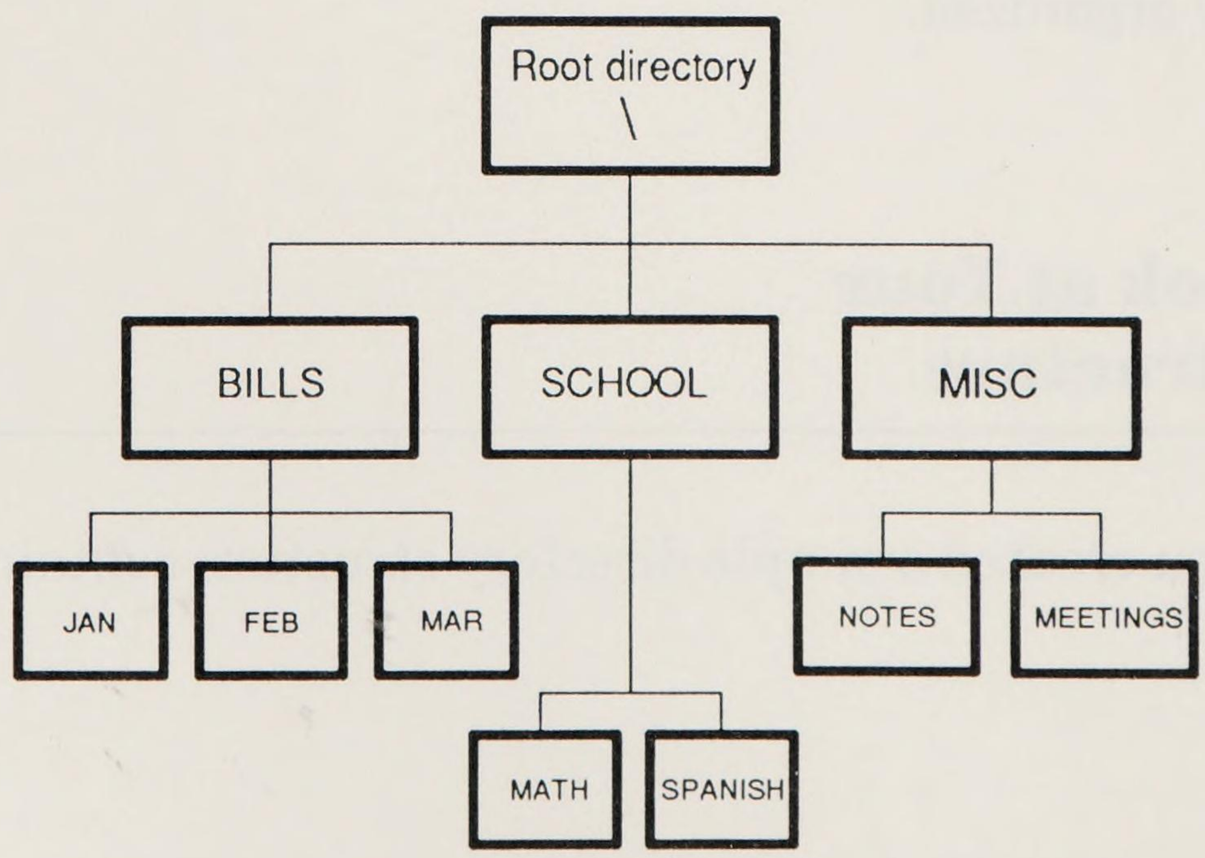


Figure 14-1. A more complex directory structure

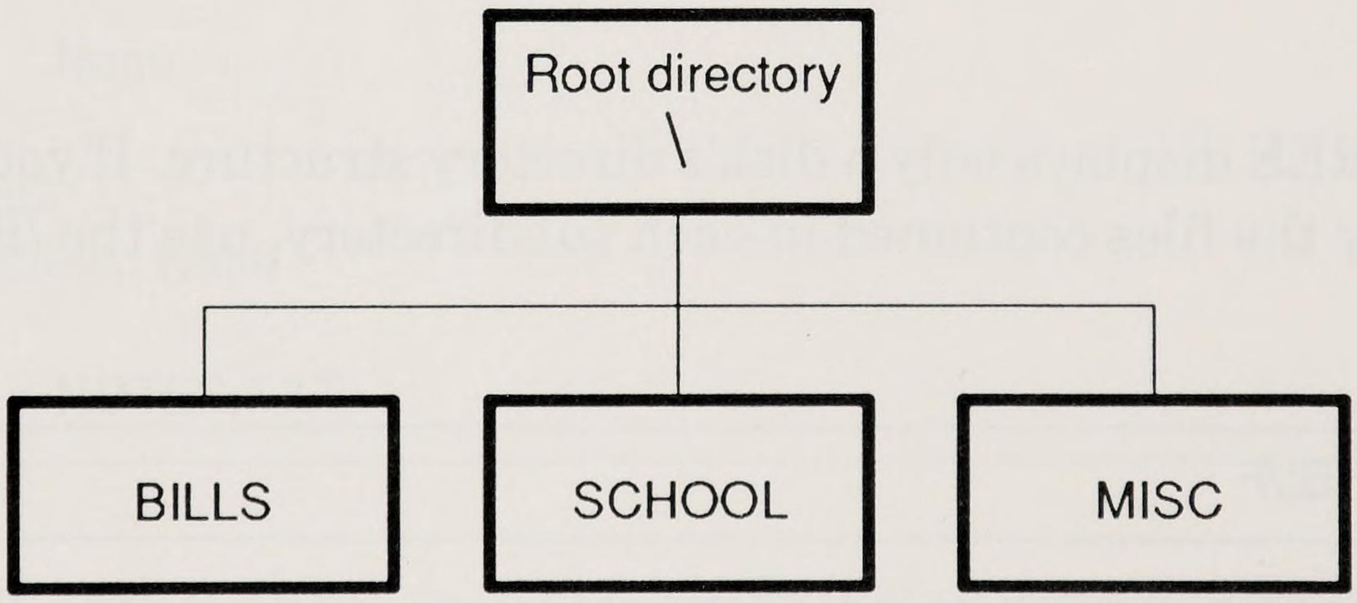
Instead, DOS provides the TREE command, which displays a disk's directory structure.

Tip: *The DOS TREE command displays a disk's directory structure on your screen. Optionally, TREE displays all of the files in each subdirectory on the disk.*

With the Supplemental Programs Disk in drive A, issue the command

```
A> DIR  TREE.COM

          Volume in drive A is MS330PP02
          Directory of  A:\

TREE      COM     3540     7-24-87  12:00a
          1 File(s)  2048 bytes free
```

As you can see, TREE is an external DOS command. If you place the disk containing



in drive B, the command

```
A> TREE  B:
```

will display the disk's directory structure, as shown here.

```
A> TREE  B:

DIRECTORY PATH LISTING

Path: \BILLS
Subdirectories:  None


Path: \SCHOOL
Sub-directories: None


Path: \MISC
Sub-directories: None
```

If you want to obtain a hardcopy listing of the directory structure, use the output redirection operator, as shown here:

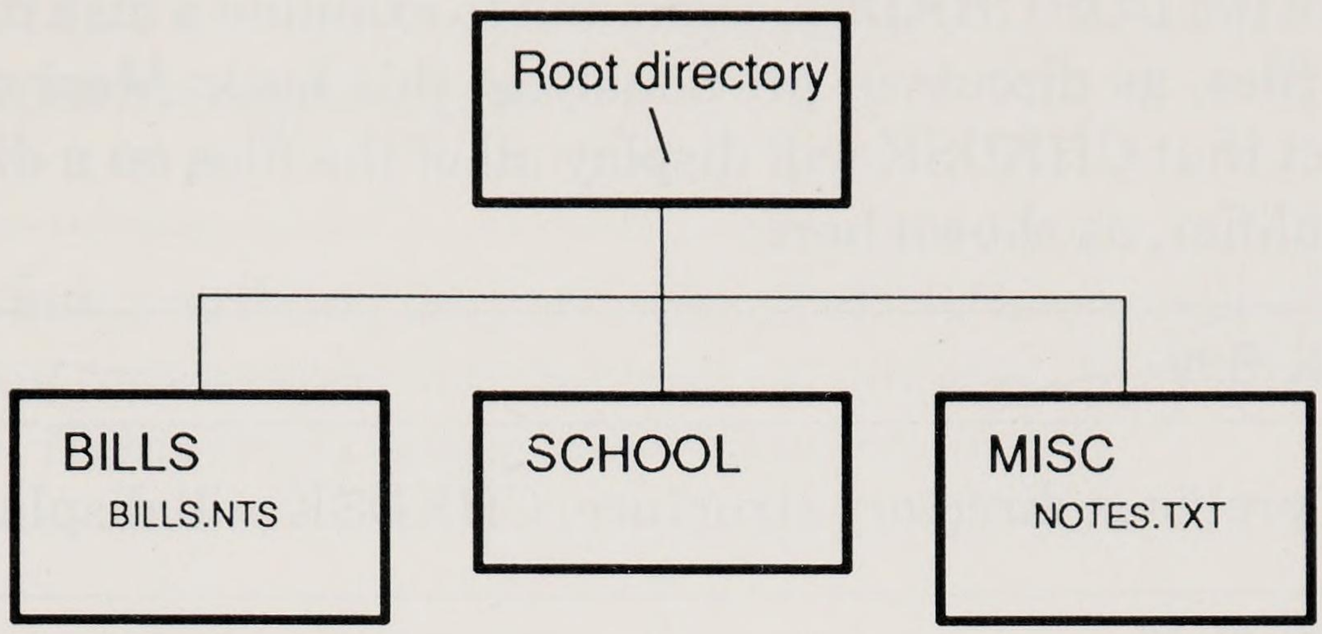
```
A> TREE  B: > PRN
```

Tip: *The /F qualifier instructs TREE to display the files contained in each of the subdirectories on a disk.*

By default, TREE displays only a disk's directory structure. If you want TREE to also display the files contained in each subdirectory, use the /F qualifier, as shown here:

```
A> TREE  B:/F
```

Given the directory structure



TREE will display

```
A> TREE  B:/F

DIRECTORY PATH LISTING
Files:          None
Path: \BILLS
Sub-directories: None

Files:          BILLS.NTS

Path: \SCHOOL
Sub-directories: None

Files:          None

Path: \MISC
Sub-directories: None

Files:          NOTES.TXT
```

Tip: *The /V qualifier instructs CHKDSK to display all of the files contained on a disk.*

Most users issue the DOS CHKDSK command to examine a disk for lost sectors or fragmented files, as discussed previously in this book. Most users are not aware of the fact that CHKDSK will display all of the files on a disk if they include the /V qualifier, as shown here:

```
A> CHKDSK  B:/V
```

Given the previous directory structure, CHKDSK will display

```
A> CHKDSK  /V  B:

Directory B:\
Directory B:\BILLS
     B:\BILLS\BILLS.NTS
Directory B:\SCHOOL
Directory B:\MISC
     B:\MISC\NOTES.TXT

362496 bytes total disk space
   3072 bytes in 3 directories
   2048 bytes in 2 user files
357376 bytes available on disk

655360 bytes total memory
587488 bytes free
```

As with the DOS TREE command, you can redirect the output of the CHKDSK command to your printer.

```
A> CHKDSK  B:  /V  >  PRN
```

Using PATH to Locate DOS Command Files

This book previously discussed the DOS environment and the PATH= entry, as shown here.

```
A> SET
COMSPEC=A:\COMMAND.COM
PATH=
```

At that time, you found that each time DOS fails to locate an external command in the current or specified directory, it searches the environment for the PATH= entry. If this entry specifies additional locations

```
A> SET
COMSPEC=A:\COMMAND.COM
PATH=A:;B:
```

DOS will search each location until it finds the specified command or batch file, or until it exhausts the entries in the command path. If DOS fails to locate an external command, it searches the current directories of the disks in drives A and B, in that order.

In Chapter 13 you learned that DOS allows you to place files containing external DOS commands into DOS subdirectories. To execute those files, you precede the command's name with a subdirectory path that contains the command, as shown here:

```
A> \DOS\FORMAT B:
```

Fixed-disk users should place all of their commands in DOS subdirectories. Once you do so, you will probably want to include several of the directories that contain your commonly used commands in the DOS command search path. For example, to place your DOS subdirectory in the current command search path, use the command

```
A> \DOS\PATH A:\DOS
```

Tip: *PATH is an internal DOS command that allows you to define the command search path associated with the PATH= entry in the DOS environment.*

The format of the PATH command is

```
PATH [drive:][path][; ...]
```

or

```
PATH ;
```

The three periods in the command format state that PATH allows you to specify multiple PATH entries, each separated by a semicolon, as shown in this fixed-disk example.

```
C> PATH C:\DOS;C:\LOTUS;C:\WORDSTAR
```

In this case, if DOS fails to find an external command as specified in the command line, it will search the DOS, LOTUS, and WORDSTAR subdirectories in that order for the command.

As you will see later in this chapter, at a minimum, fixed-disk users will want to include in their command search path the subdirectory that contains their DOS external commands. For now, issue the command

```
A> PATH A:\DOS
```

Using the DOS SET command to examine the environment reveals

```
A> SET
COMSPEC=A:\COMMAND.COM
PATH=A:\DOS
```

As you can see, DOS has updated the PATH= entry to reflect the desired command search path. To better understand this processing, take a look at an example. Place a newly formatted disk in drive B. Using MKDIR, create the DOS subdirectory on that disk, as shown here:

```
A> MKDIR B:\DOS
```

Next, copy the LABEL.COM file from your Program Disk to the DOS subdirectory on drive B.

```
A> COPY  LABEL.COM  B:\DOS
        1 File(s) copied
```

Insert the disk from drive B into drive A and list the contents of the root directory.

```
A> DIR

          Volume in drive A has no label
          Directory of  A:\

DOS       <DIR>   7-02-88  7:27a
          1 File(s)      358400 bytes free
```

As you can see, the LABEL.COM file does not reside in the root. However, if you issue the command

```
A> LABEL
```

DOS invokes the program and displays

```
Volume in drive A has no label

Volume label (11 characters, ENTER for none)?
```

In this case, either type a disk name or use the CTRL- BREAK key combination to terminate the command. When DOS failed to locate the LABEL.COM file in the current directory, it searched the DOS environment for the PATH= entry. DOS next searched the DOS subdirectory for the command as specified in the command search path. Therefore, although the LABEL command did not reside in the current directory, by using the command search path, DOS was able to find it.

Tip: *Place the subdirectories most likely to contain your external commands first in the command path.*

Assume, for example, that your command search path contains

```
A> SET
COMSPEC=A:\COMMAND.COM
PATH=C:\DATA;C:\MISC;C:\DOS
```

Each time that DOS fails to locate an external command, it begins searching the subdirectories in the specified ordered. If the first two subdirectories contain mostly data files, DOS is needlessly examining those files in search of your commands. These additional file-search operations are quite time-consuming, adding unnecessary system overhead.

Tip: *To display the current command search path, invoke the DOS PATH command without any command-line parameters.*

If you simply enter **PATH** at the DOS prompt

```
A> PATH
```

DOS will display the current command path, as shown here:

```
A> PATH
PATH=C:\DATA;C:\MISC;C:\DOS
```

Tip: *If a subdirectory in the command search path is invalid, DOS ignores that path entry.*

When you invoke the DOS PATH command, DOS does not validate the subdirectories that you specify, ensuring that each resides on disk. Therefore, if you misspell a subdirectory name, PATH does not care.

Tip: *Include disk drive specifiers whenever possible in your command search path entries.*

Consider this command search path.

```
A> SET
COMSPEC=A:\COMMAND.COM
PATH=\DOS;\LOTUS;\WORDSTAR
```

As before, if DOS fails to locate an external command, it will search the DOS, LOTUS, and WORDSTAR subdirectories. However, if you change your default disk drive, you will also change the disk drive on which DOS will search for the directories. If each path entry instead contains a disk drive specifier

```
A> SET
COMSPEC=A:\COMMAND.COM
PATH=C:\DOS;C:\LOTUS;C:\WORDSTAR
```

DOS will always search for the command on the correct disk.

Tip: *To remove the current command search path, invoke PATH with only a semicolon in the command line.*

If you remove the command search path, DOS will simply search for commands in the current directory or specified directory. If DOS fails to locate the command, it will display the familiar message

```
Bad command or file name
```

To remove a command search path, invoke PATH with a semicolon, as shown here:

```
A> PATH ;
```

Using SET to display the environment entries now reveals

```
COMSPEC=A:\COMMAND.COM
PATH=
```

Likewise, if you simply enter the PATH command, DOS will display

```
No Path
```

Tip: *Once you determine the correct PATH command for your directory structure, place the PATH command in the AUTOEXEC.BAT file.*

If you are using a fixed disk, identify the subdirectories that you want to place in the command search path and issue a corresponding PATH command in AUTOEXEC.BAT. This will in turn ensure that your system has the same command search path available each time it starts.

Installing DOS on Your Fixed Disk

This section will get fixed-disk users up and running with DOS. If your system is floppy-disk based, you may still want to read this section to see how easy it is to install DOS on a fixed disk. Chapter 4 discussed several of the steps that you must perform to prepare your fixed disk for use by DOS. If your fixed disk currently will not boot DOS when drive A does not contain a floppy disk, refer to the steps discussed in Chapter 4. If DOS will start from your fixed disk, you need to ensure that DOS is correctly installed on the disk.

All of your DOS external commands should reside in a DOS subdirectory. Most users name this directory *DOS*. Create a subdirectory for your external commands if you have not done so already.

```
C> MKDIR \DOS
```

Tip: *All DOS external commands should reside in distinct subdirectories on your fixed disk.*

Place your Program Disk in drive A and copy the files that it contains to the DOS subdirectory on your fixed disk.

```
C> COPY A:*.* \DOS
```

Repeat this process for the Supplemental Programs Disk.

```
A> COPY  A:*.*  \DOS
```

Next, perform a DOS directory listing of your root directory.

```
C> DIR

                Volume in drive C has no label
                Directory of  C:\

DOS             <DIR>   7-02-88  7:33a
                1 File(s) 20135424 bytes free
```

If your root directory contains files other than

AUTOEXEC.BAT
COMMAND.COM
CONFIG.SYS

you will need to place the other files into corresponding subdirectories on the disk. If your disk does not contain the CONFIG.SYS file

```
C> DIR  CONFIG.SYS

File not found
```

create one, as shown here:

```
C> COPY  CON  CONFIG.SYS
BUFFERS=25
FILES=20
^Z
                1 File(s) copied
```

Finally, copy the COMMAND.COM file from the Program Disk in drive A to the root directory on drive C.

```
C> COPY  A:COMMAND.COM  C:\
                1 File(s) copied
```

Ensure that you can still boot DOS from your fixed disk by using the CTRL-ALT-DEL key combination.

If your system displays the message

```
Non-system disk or disk error
Replace and strike any key when ready
```

DOS was not properly installed on your fixed disk (as discussed in Chapter 4) by the

```
A> FORMAT C: /S
```

command. Likewise, if your system displays the message

```
Bad or missing Command Interpreter
```

you must copy the COMMAND.COM file to the root directory of drive C, as just discussed. In most cases, if you have followed the steps in Chapter 4 and this chapter, your system will boot from drive C with no problems.

Most users will want to issue the command

```
C> COPY CON AUTOEXEC.BAT
PATH C:\DOS
^Z
            1 File(s) copied
```

creating the desired command path each time the system starts.

Creating a Data-File Search Path

Tip: *If you are using DOS version 3.3, DOS provides the APPEND command, which allows you to define a list of subdirectories that DOS will examine in search of your data files.*

Just as the DOS PATH command defines the list of subdirectories that DOS will search for your external commands, the DOS version 3.3 APPEND command defines the list of subdirectories that DOS will search for data files. Assume, for example, that your disk contains the subdirectories BILLS, SCHOOL, and MISC. Select the root as your current directory.

```
A> CD \
```

The command

```
A> TYPE BILLS.NTS
File not found

A>
```

fails because DOS cannot find the BILLS.NTS file in the current directory. However, if you include BILLS in the data-file search path

```
A> APPEND A:\BILLS
```

the TYPE command now results in

```
A> TYPE BILLS.NTS
The subdirectory BILLS contains
my monthly expenses.

A>
```

The command succeeds because DOS now searches each directory in the data-file search path. To include all three of the BILLS, SCHOOL, and MISC directories your APPEND command would become

```
A> APPEND A:\BILLS;A:\MISC;A:\SCHOOL
```

Just as with the PATH command, if you simply type **APPEND** and press ENTER, DOS will display your current data-file search path.

```
A> APPEND
APPEND=A:\BILLS;A:\MISC;A:\MEETINGS
```

The complete format for the APPEND command now becomes

```
APPEND   [/X][/E]   [drive:][path][;...]
```

or

```
APPEND ;
```

As before, the three periods in the command format tell you that APPEND allows you to place multiple path names in the APPEND data-file search path. To delete your data-file search path, type **APPEND** followed immediately by a semicolon.

```
C> APPEND ;
```

Trap: *By default, not all DOS commands recognize the data- file search path.*

To expand the number of applications that use the data-file search path, the *first time* that you invoke APPEND you must use the /X qualifier, as shown here:

```
A> APPEND /X
```

Unlike the DOS PATH command, APPEND does not place an entry in your DOS environment by default.

```
C> SET
COMSPEC=C:\COMMAND.COM
PATH=C:\DOS;
```

To do so (although it is not required for the command to work), use the /E qualifier the first time that you invoke APPEND, as follows:

```
A> APPEND /E
```

Some people are concerned that if they create a data-file search path, DOS might inadvertently delete a file when given a command such as

```
C> DEL FILENAME.EXT
```

However, DOS has already taken this into account. If you attempt to delete a file that does not reside in the current directory or specified directory, and DOS locates that file in the data-file search path, DOS will display the message

```
Access Denied
```

instead of deleting the file. In this way, your files are indeed protected from errant commands.

If you have a subdirectory of data files that you use on a regular basis, you may want to include the subdirectory in a data-file search path. Remember, though, that if you place too many directories in the data-file search path, DOS will spend more time searching through files than performing useful tasks.

Protecting Your Files With Unique Subdirectory Names

One of the problems that arises with personal computers is that a person who gains access to your computer, in most cases, also gains access to your files by performing standard DOS file and subdirectory commands.

Luckily, however, not all users have the same level of expertise. Therefore, you can use a little-known DOS fact to protect your files from most users. In Chapter 6 you learned that the ALT key allows you to easily access the extended character set. If you have a subdirectory that you don't want other users to access, include an extended character in the directory name.

```
A> DIR

Volume in drive A has no label
Directory of  A:\

DOS        <DIR>        7-02-88  7:27a
SÄLARY     <DIR>        7-02-88  7:53a
           2 File(s)    357376 bytes free
```

Note the unique second letter in the directory name SÄLARY.

To create the directory name shown, first type **MKDIR** and the letter **S**:

```
A> MKDIR  S
```

Next, hold down the ALT key and use the numeric keypad at the right of your keyboard to enter the number **142**. When you release the ALT key, DOS will display

```
A> MKDIR  SÄ
```

Next, type the remaining letters of the directory name.

```
A> MKDIR  SÄLARY
```

To access this directory, use the same technique.

```
A> CHDIR  SÄLARY
```

Since nine out of ten users do not know that you can include an extended character in the directory name in this manner, they are unlikely to be able to access your files.

Tip: *Using characters from the extended character set in your DOS directory names will prevent most users from knowing how to access your files.*

Upgrading DOS

Upgrading DOS to a newer version differs from installing DOS. If you are upgrading, you already have files residing on your disk. Before you upgrade, organize the files on your disk, placing them into subdirectories. When the only files that remain in your root directory are

```
AUTOEXEC.BAT
COMMAND.COM
CONFIG.SYS
```

restart your computer from a floppy disk using the newer version of DOS.

Tip: *The DOS SYS command transfers to your target disk the hidden system files that DOS requires to boot. SYS is an external DOS command.*

With your Program Disk in drive A, issue the command

```
A> SYS  C:
```

If the SYS command is successful, DOS will display the message

```
System transferred
```

Your next step in this case is to copy the COMMAND.COM file from your new Program Disk to the root directory of drive C. Then copy the files on the Program Disk and Supplemental Programs Disk to the subdirectory containing your DOS commands.

Trap: *The SYS command cannot always upgrade your disk as desired. If SYS fails, DOS will display*

```
No room for the system on destination disk
```

If you cannot successfully transfer the new version of DOS to your disk using the DOS SYS command, you will have to perform a backup-and-restore operation, reformatting your disk as discussed in the chapter on backing up your disks.

Tips and Traps Summary

Taking a Look at Your Directory Structure

Tip: The DOS TREE command displays a disk's directory structure on your screen. Optionally, TREE displays all of the files in each subdirectory on the disk.

Tip: The /F qualifier instructs TREE to display the files contained in each of the subdirectories on a disk.

Tip: The /V qualifier instructs CHKDSK to display all of the files contained on a disk.

Using PATH to Locate DOS Command Files

Tip: PATH is an internal DOS command that allows you to define the command search path associated with the PATH= entry in the DOS environment.

Tip: Place the subdirectories most likely to contain your external commands first in the command path.

Tip: To display the current command search path, invoke the DOS PATH command without any command-line parameters.

Tip: If a subdirectory in the command search path is invalid, DOS ignores that path entry.

Tip: Include disk drive specifiers whenever possible in your command search path entries.

Tip: To remove the current command search path, invoke PATH with only a semicolon in the command line.

Tip: Once you determine the correct PATH command for your directory structure, place the PATH command in your AUTOEXEC.BAT file.

Tips and Traps Summary (*continued*)

Installing DOS on Your Fixed Disk

Tip: All DOS external commands should reside in distinct subdirectories on your fixed disk.

Creating a Data-File Search Path

Tip: If you are using DOS version 3.3, DOS provides the APPEND command, which allows you to define a list of subdirectories that DOS will examine in search of your data files.

Trap: By default, not all DOS commands recognize the data-file search path.

Protecting Your Files with Unique Subdirectory Names

Tip: Using characters from the extended character set in your DOS directory names will prevent most users from knowing how to access your files.

Upgrading DOS

Tip: The DOS SYS command transfers to your target disk the hidden system files that DOS requires to boot. SYS is an external DOS command.

Trap: The SYS command cannot always upgrade your disk as desired. If SYS fails, DOS will display

```
No room for the system on destination disk
```

15

Using Logical Disk Drives and DOS Pretender Commands

A physical device is a hardware device that is actually connected to your computer, such as a printer or disk drive. Since most users have different hardware configurations, the manner in which they have executed the DOS commands throughout this book has differed. For example, those of you with dual-floppy disk drive systems have issued commands such as

```
A> DISKCOPY A: B:
A> TREE B: /F
A> LABEL B:
```

with your Program Disk in drive A and your data disk in drive B. Up to now, your second floppy-disk drive has proved to be quite convenient. However, this chapter teaches you several techniques that you can use to trick DOS into believing that your computer has more disk drives than are physically connected to it. Because these disk drives do not actually exist, many people classify the DOS commands that will be examined here as *pretender commands.* Likewise, the

disk drives in this case are not physical disk drives connected to your system, but *logical disk drives.*

As you will see in this chapter, regardless of the actual number of floppy-disk drives on your system, DOS always allows you to refer to drives A and B.

Using Drives A and B In a Single-Floppy-Disk System

If your system has only one floppy-disk drive, you have been accessing the files contained on the disk in that drive by using the disk drive specifier A.

```
A> DIR
```

If this is the case, place your Program Disk in drive A and issue the command

```
A> DIR  B:
```

DOS will respond with the message

```
Insert diskette for drive B: and strike
any key when ready
```

Place the disk that you to want to examine into the drive that you have been referring to as drive A, and press ENTER. DOS will display the directory listing of that disk, as shown here:

```
A> DIR  B:

Insert diskette for drive B: and strike
any key when ready

Volume in drive B is MS330PP02
Directory of  B:\

BACKUP        COM        29976        7-24-87  12:00a
```

```
DEBUG      COM      15866      7-24-87  12:00a
EGA        CPI      49065      7-24-87  12:00a
GWBASIC    EXE      80592      7-24-87  12:00a
KEYBOARD   SYS      19735      7-24-87  12:00a
LCD        CPI      10752      7-24-87  12:00a
LINK       EXE      39172      7-24-87  12:00a
PRINTER    SYS      13559      7-24-87  12:00a
RAMDRIVE   SYS       6481      7-24-87  12:00a7
REPLACE    EXE      13234      7-24-87  12:00a
RESTORE    COM      35650      7-24-87  12:00a
SHARE      EXE       8608      7-24-87  12:00a
TREE       COM       3540      7-24-87  12:00a
XCOPY      EXE      11216      7-24-87  12:00a
FC         EXE      15974      7-24-87  12:00a
        15 File(s)    2048 bytes free
```

Note that, to DOS, this disk is now contained in drive B.

```
Volume in drive B is MS330PP02
Directory of  B:\
```

When the directory listing completes, DOS will prompt you to place the original disk that you had in drive A back into the drive.

```
Insert diskette for drive A: and strike
any key when ready
```

Do so, and press ENTER. As you just saw, your single- floppy disk drive system supports two logical disk drives, A and B. When you reference either drive A or drive B, DOS will logically differentiate between the two.

Tip: *If your system has only one physical disk drive, DOS allows you to reference two logical disk drives, A and B.*

With your Program Disk in drive A and a formatted, unused disk available, issue the command

```
A> COPY ANSI.SYS B:
```

DOS will again respond with

```
Insert diskette for drive B: and strike
any key when ready
```

Place the new disk in the disk drive and press ENTER. Once DOS copies the file to this disk, it will prompt you to reinsert the original disk into drive A.

```
        1 File(s) copied

Insert diskette for drive A: and strike
any key when ready
```

Do so, and press ENTER. Remove your Program Disk from drive A and insert the second disk. A directory listing of the disk reveals

```
A> DIR

Volume in drive A has no label
Directory of  A:\

ANSI        SYS        1678        3-17-87  12:00p
            1 File(s) 360448 bytes free
```

As you can see, DOS still considers this disk to reside in drive A:

```
Volume in drive A has no label
Directory of  A:\
```

When you use logical disk drives, DOS maintains the drive letter of the last default drive. Prior to the COPY command, drive A was the default. When the COPY command completed, as you can see by the DOS prompt, drive A remained the default disk drive. Although you created the second disk in drive B, to DOS, you placed a new floppy disk in drive A.

By using logical disk drives in this manner, single-floppy disk drive systems can fully support commands such as

```
A> DISKCOPY A: B:
A> LABEL B:
A> COPY *.* B:
A> COMP A:FILE.EXT B:FILENAME.EXT
A> DIR > B:FILE.EXT
```

Watch, for example, how DOS performs the following DISKCOPY command on a single-floppy disk drive system.

```
A> DISKCOPY A: B:
```

DOS displays

```
Insert SOURCE diskette in drive A:

Press any key when ready . . .

Copying 40 tracks
9 Sectors/Track, 2 Side(s)

Insert TARGET diskette in drive A:

Press any key when ready . . .

Copy another diskette (Y/N)?
```

Although it may take a little longer, due to the exchanging of floppy disks in the drive, a single-floppy disk system can do everything a dual-floppy disk system can do.

Making One Disk Drive Appear as Another

Just as DOS can make your system appear to have more disk drives than are actually connected to it, so too can it make your system appear to have less.

Some older software packages always look for specific files on the floppy disk in drive A. A problem arises when fixed-disk users want to place the program into a DOS subdirectory on their fixed disk. If the program continually looks to the floppy disk in drive A for files, you have really not gained anything by placing the program on the fixed disk.

Tip: *The DOS ASSIGN command allows you to direct DOS disk drive references from one disk drive to another.*

Assume, for example, that you have just placed a DOS program on your fixed disk that always looks to drive A for files, as just discussed. Before you invoke this program, issue the command

```
C> ASSIGN C=A
```

In this case, DOS will route future disk drive references for drive A to drive C. Therefore, when the program later looks for files on drive A, DOS will route the request to drive C, where the files reside.

If you have a dual-floppy disk system, issue the command

```
A> ASSIGN A=B
```

If you have a fixed-disk drive, issue the command

```
C> ASSIGN C=A
```

In the first case (A=B), DOS will route all disk references for drive B to drive A. The command

```
A> DIR B:
```

actually displays a directory listing of the files on drive A. Likewise, with the fixed-disk drive assignment (C=A), DOS will route all of the disk drive references for drive A to drive C.

Tip: *To cancel disk drive reassignment, enter the DOS ASSIGN command without any command-line parameters.*

To resume normal disk drive operations following an ASSIGN command, simply invoke ASSIGN as

```
A> ASSIGN
```

with no command-line parameters.

Tip: *To maintain compatibility with future versions of DOS, use the DOS SUBST command in place of ASSIGN.*

Later on, this chapter will discuss the DOS SUBST command. Microsoft recommends that you use SUBST instead of ASSIGN whenever possible.

Making Your Disks Appear Larger than They Are

Although it is rare, there may be instances when a database application needs to store more information on a floppy disk than the disk actually has space for. In some cases, the DOS JOIN command allows you to make two small disks appear as one large disk.

Tip: *The DOS JOIN command instructs DOS to combine (join together) two disks, making them appear to be one large disk.*

For this example you will join the two floppy-disk drives A and B, creating a larger, logical disk in drive B. JOIN is an external command.

```
A> DIR JOIN.EXT

Volume in drive A is MS330PP01
Directory of A:\

JOIN        EXE        9612        7-24-87  12:00a
            1 File(s)  5120 bytes free
```

Place your Program Disk in drive A. Next, JOIN requires you to create an empty directory on the target disk.

```
B> MKDIR \JOINDIR
```

To create the join, issue the command

```
B> A:JOIN A: B:\JOINDIR
```

Next, select JOINDIR as your current directory, as shown here:

```
B> CD \JOINDIR
```

Create the following file:

```
B> COPY CON JOIN.TXT
The DOS JOIN command directs DOS
to make two smaller disks appear
as one large disk.
^Z
            1 File(s) copied

B>
```

When you press ENTER, DOS will actually write the file's contents to the disk in drive A. Since DOS has joined drives A and B together, it will first use the disk with sufficient disk space for the file copy. If you issue the DOS DIR command, DIR not only displays the JOIN.TXT file that you just created but all of the files on the disk in drive A. If you insert a newly formatted disk in drive A, your applications will essentially have two disks worth of storage space associated solely with drive B.

Trap: *Once you join a disk drive to a DOS subdirectory, you can no longer reference that drive.*

In the case of the previous example, DOS has joined drive A to the subdirectory B:\JOINDIR. If you attempt to access drive A

```
B> DIR A:
```

DOS will display

```
Invalid drive specification
```

As long as drive A is joined to the subdirectory on drive B, DOS will not allow you to access it.

Tip: *If you invoke JOIN without specifying command-line parameters, JOIN will display the current JOIN assignment.*

In this case, if you issue the command

```
B> JOIN
```

DOS will display

```
A: => B:\ JOINDIR
```

Tip: *The /D qualifier instructs JOIN to remove a disk drive join.*

Once you no longer require the disk drive join, place the /D qualifier immediately after the joined disk drive specifier, as shown here:

```
B> JOIN A: /D
```

DOS can again access drive A, and the directory of the files in the subdirectory B:\JOINDIR no longer displays the files that reside on drive A.

Abbreviating DOS Subdirectory Names

If you use DOS subdirectories to a great extent to increase your file organization, there may be times when your directory names become quite large, as shown in Figure 15-1.

If you have to use one or two of these directories on a regular basis, typing the subdirectory name can become quite time-consuming.

Tip: *The DOS SUBST command allows you to abbreviate your directory names as a logical-disk drive identifier.*

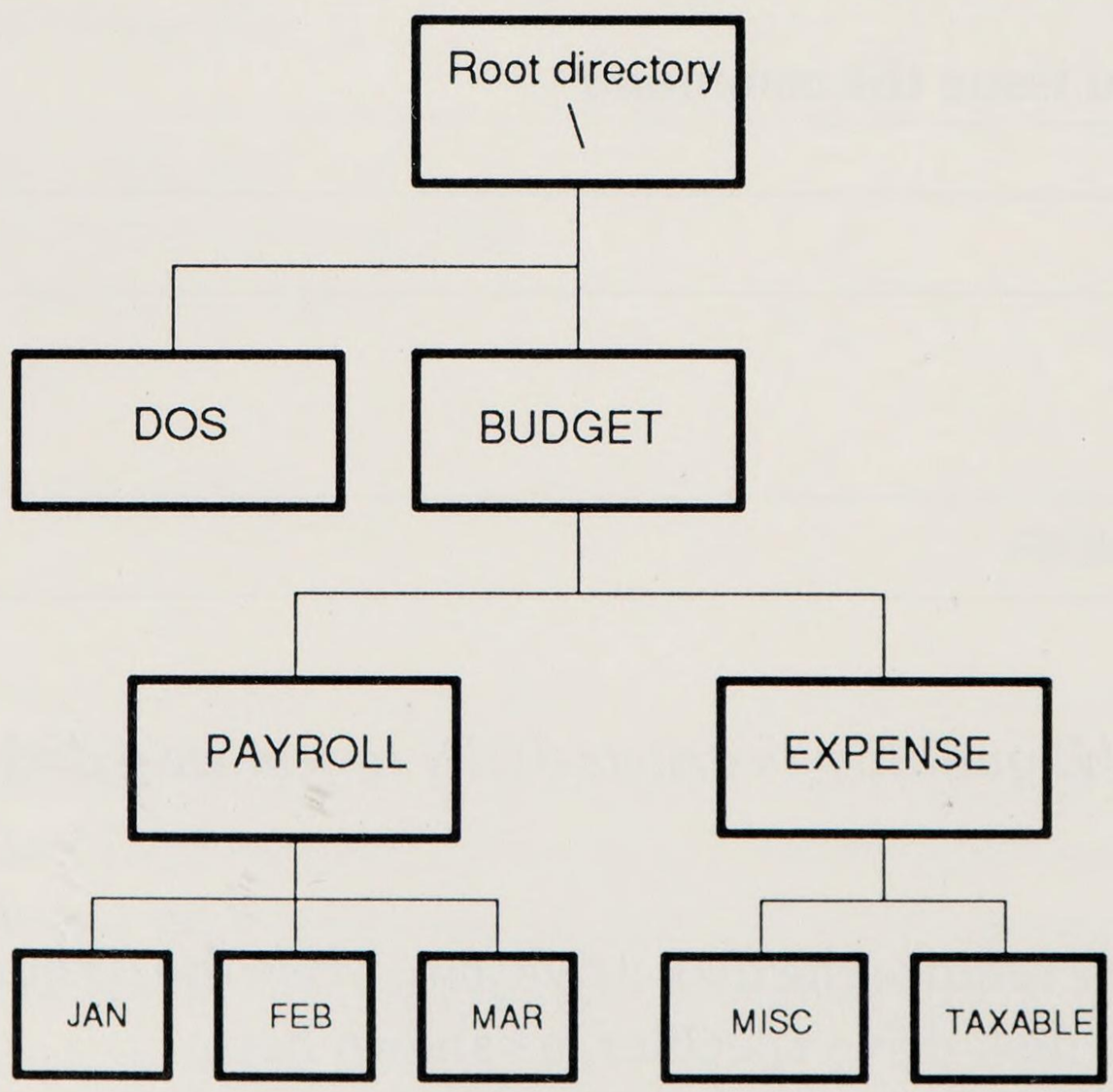


Figure 15-1. A directory with a large number of subdirectories

Assume that you must access the files in the subdirectory \BUDGET\PAYROLL\JUNE on a regular basis.

```
C> TYPE \BUDGET\PAYROLL\JUNE\JUNE.EXP
```

or

```
C> CD \BUDGET\PAYROLL\ JUNE
```

Rather than having to repeatedly type this long directory name, DOS allows you to substitute the directory name with a logical disk drive identifier, as shown here:

```
C> SUBST D: C:\BUDGET\PAYROLL\JUNE
```

Once you do so, DOS allows you to use the drive letter D, in this case, in place of the subdirectory path. The command

```
C> TYPE \BUDGET\PAYROLL\ JUNE\ JUNE.EXP
```

becomes

```
C> TYPE D:JUNE.EXP
```

and the command

```
C> CD \BUDGET\PAYROLL\ JUNE
```

becomes

```
C> CD D:
```

If the subdirectory that you have abbreviated contains other DOS subdirectories, you can still refer to them by using drive D, as shown here:

```
C> CD D:\SUBDIR
```

When DOS encounters the logical disk drive D, it substitutes in the corresponding path name, yielding

```
C> CD  C:\BUDGET \PAYROLL\ JUNE\SUBDIR
```

Tip: *If you invoke the SUBST command without any command-line parameters, DOS will display the current substitutions.*

In this case, to display the directory abbreviation, enter

```
C> SUBST
```

DOS will display

```
D: => C:\BUDGET \PAYROLL\ JUNE
```

Tip: *The /D qualifier instructs SUBST to remove a directory substitution.*

To remove a directory substitution, place the /D qualifier in the SUBST command line immediately after the logical-disk drive specifier, as shown here:

```
C> SUBST  D:  /D
```

Trap: *By default, DOS supports only logical disk drives A through E.*

If you plan on performing several directory-name substitutions, you will very likely have to modify the LASTDRIVE= entry in CONFIG.SYS. As discussed in Chapter 9, the LASTDRIVE= entry specifies the logical-disk drive letter, A-Z, of the last logical disk drive DOS will support. To support the logical drives A-I, for example, you must use

```
LASTDRIVE=I
```

Remember that once you modify a CONFIG.SYS entry, you must reboot DOS for the change to take effect.

A Word of Warning

As you have just seen, DOS logical-disk and pretender commands allow you to trick DOS into believing that your system has more disk drives available than are actually connected to it. Although such processing can be very convenient, it can also be very destructive. You must be careful which DOS commands you issue in conjunction with DOS logical disks. For example, do not use the DOS DISKCOPY, DISKCOMP, FORMAT, LABEL, or RESTORE command with a logical disk drive. If you do, DOS may not actually access the disk that you intended, perhaps losing the data your disk contains.

All of the DOS commands that should not be used in conjunction with DOS logical disk drives contain a warning message, as you will see in the command reference section (Appendix B) of this book.

Tips and Traps Summary

Using Drives A and B in a Single-Floppy-Disk System

Tip: If your system has only one physical disk drive, DOS allows you to reference two logical disk drives, A and B.

Making One Disk Drive Appear as Another

Tip: The DOS ASSIGN command allows you to direct DOS disk drive references from one disk drive to another.

Tip: To cancel disk drive reassignment, enter the DOS ASSIGN command without any command-line parameters.

Tip: To maintain compatibility with future versions of DOS, use the DOS SUBST command in place of ASSIGN.

Making Your Disks Appear Larger than They Are

Tip: The DOS JOIN command instructs DOS to combine (join together) two disks, making them appear to be one large disk.

Trap: Once you join a disk drive to a DOS subdirectory, you can no longer reference that drive.

Tip: If you invoke JOIN without specifying command-line parameters, JOIN will display the current JOIN assignment.

Tip: The /D qualifier instructs JOIN to remove a disk drive join.

Tips and Traps Summary (*continued*)

Abbreviating DOS Subdirectory Names

Tip: The DOS SUBST command allows you to abbreviate your directory names as a logical-disk drive identifier.

Tip: If you invoke the SUBST command without any command-line parameters, DOS will display the current substitutions.

Tip: The /D qualifier instructs SUBST to remove a directory substitution.

Trap: By default, DOS supports only logical disk drives A through E.

16

Getting the Most from the DOS File-Manipulation Commands

Throughout this book you have learned to create, rename, delete, copy, print, display, and organize your files by placing them into DOS subdirectories. Perhaps the most important function that computers provide is information storage and retrieval. This chapter looks at the last set of DOS file-manipulation commands, which provide the following functionality:

FIND	Searches one or more files for a specific word or phrase
RECOVER	Recovers portions of a damaged file or disk
ATTRIB	Sets or displays a file's read-only and archive required attribute bits
COMP	Compares two files, displaying the first ten differences between the files in hexadecimal
FC	Provides an enhanced file-comparison utility for DOS version 3.3

REPLACE	Provides an easy means of file replacement and updates

This chapter will strengthen your understanding of DOS file manipulation. Take time to issue each of the examples given.

Revisiting the FIND Command

When DOS I/O redirection was examined earlier in this book, you learned that the DOS FIND command allows you to search redirected input for a specific word or phrase.

```
A> DIR | FIND "<DIR>"
```

If you examine the format of the DOS FIND command, however, you will discover that FIND supports optional file names in the command line.

```
[drive:][path]FIND [/C][/N][/V] "word or phrase" [file ...]
```

Tip: *The DOS FIND command allows you to include one or more file names in the command line. If present, FIND will examine each of the specified files one at a time for the given word or phrase.*

Create the following files.

```
A> COPY  CON  C.SW
Turbo C
Quick C
Microsoft C
^Z
            1 File(s) copied

A> COPY  CON  BASIC.SW
Quick BASIC
Turbo BASIC
```

```
GWBASIC
^Z
            1 File(s) copied

A> COPY  CON  PASCAL.SW
Turbo Pascal
Microsoft Pascal
^Z
            1 File(s) copied

A>
```

Next, issue the command

```
A> FIND  "Turbo"  B:C.SW  B:BASIC.SW  B:PASCAL.SW
```

FIND, in this case, will display

```
---------- B:C.SW
Turbo C

---------- B:BASIC.SW
Turbo BASIC

---------- B:PASCAL.SW
Turbo Pascal
```

Likewise, the command

```
A> FIND  /N  "Quick"  B:C.SW  B:BASIC.SW  B:PASCAL.SW
```

will display

```
---------- B:C.SW
[2]Quick C

---------- B:BASIC.SW
[1]Quick BASIC

---------- B:PASCAL.SW
```

As you can see, if you place multiple file names in the FIND command line, FIND examines each file one at a time in the order given, for the word or phrase. If you include the /C qualifier to display a count of the number of lines containing the specified word or phrase,

```
A> FIND /C "Quick" B:C.SW B:BASIC.SW B:PASCAL.SW
```

FIND displays a count for each file individually, as shown here:

```
---------- B:C.SW: 1

---------- B:BASIC.SW: 1

---------- B:PASCAL.SW: 0
```

If one of the files specified in the FIND command line does not exist, FIND continues with next the file, displaying the error message

```
FIND: File not found FILENAME.EXT
```

Recovering a Damaged File or Disk

Although it is rare if you treat your disks carefully, a disk can become damaged due to dust, smoke, fingerprints, or simply the wear and tear of everyday use. The damage in some cases will affect only a single file, while at other times it will impact the entire disk. If damage to your disk occurs, the DOS RECOVER command may be able to save a portion of the damaged disk or file.

Tip: *The DOS RECOVER command can sometimes save a portion of a damaged file or disk.*

As discussed in Chapter 4, DOS stores your files on disk by placing the information in disk sectors (see Figure 16-1). If a portion of the disk becomes

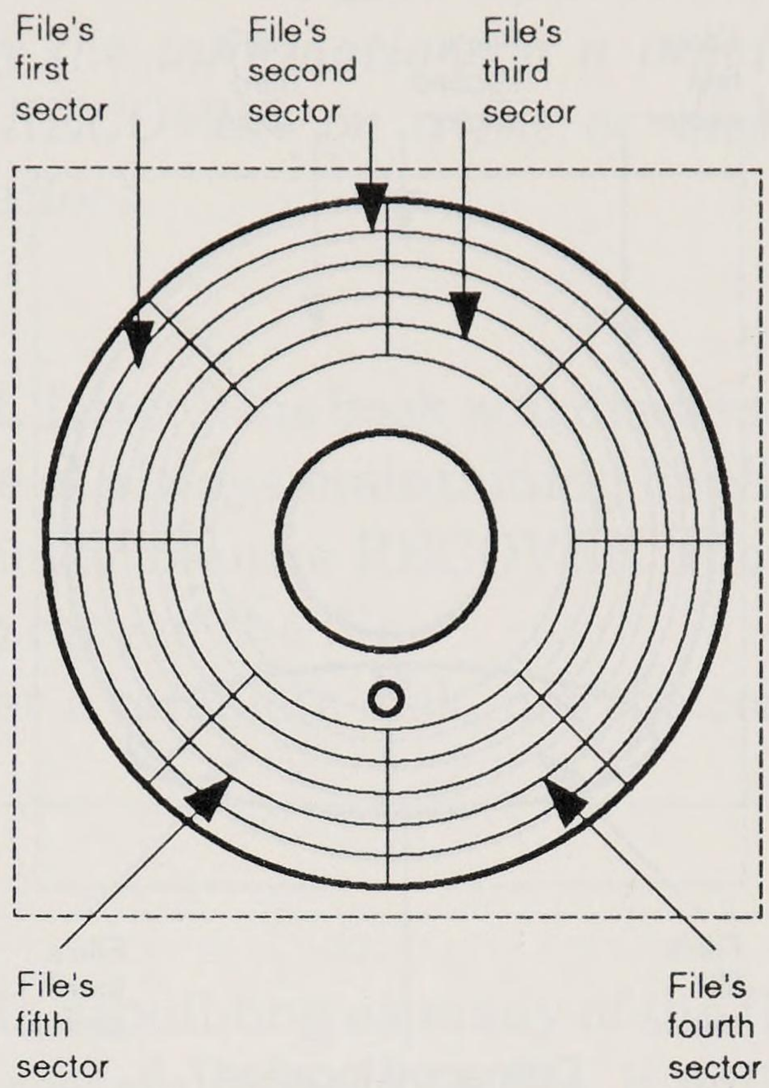


Figure 16-1. Sectors on a disk

damaged, as shown in Figure 16-2, DOS will not be able to read your entire file. In this case, a command such as

```
A> TYPE FILENAME.EXT
```

will display a portion of the file and then start displaying unrecognizable characters on your screen. If this occurs, first restart DOS and repeat the TYPE command to see if the error goes away. If it does not, the DOS RECOVER command is your last resort.

RECOVER works by rebuilding the file up to the point of the damaged location (see Figure 16-3). In this case, the recovered file would contain the first three sectors of information. You would then have to recreate the last two sectors. However, that is better than having to recreate the entire file. To recover a damaged file, invoke RECOVER with the file name in the command line, as shown here:

```
A> RECOVER B:FILENAME.EXT
```

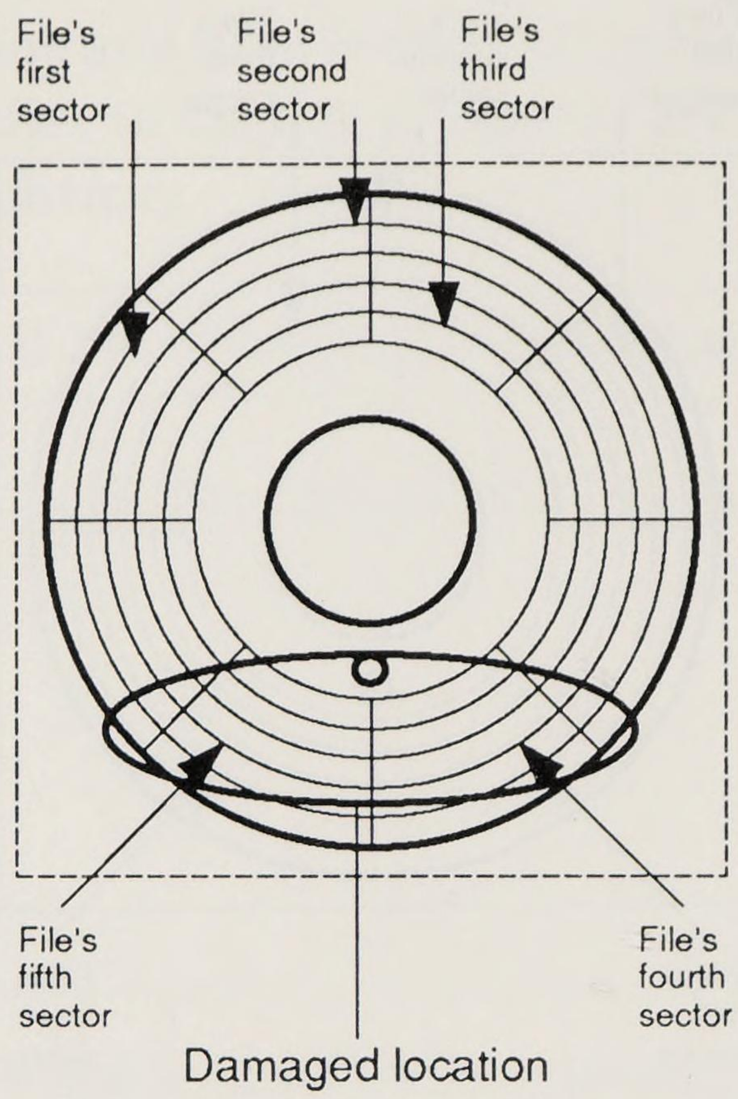


Figure 16-2. Disk with damaged location

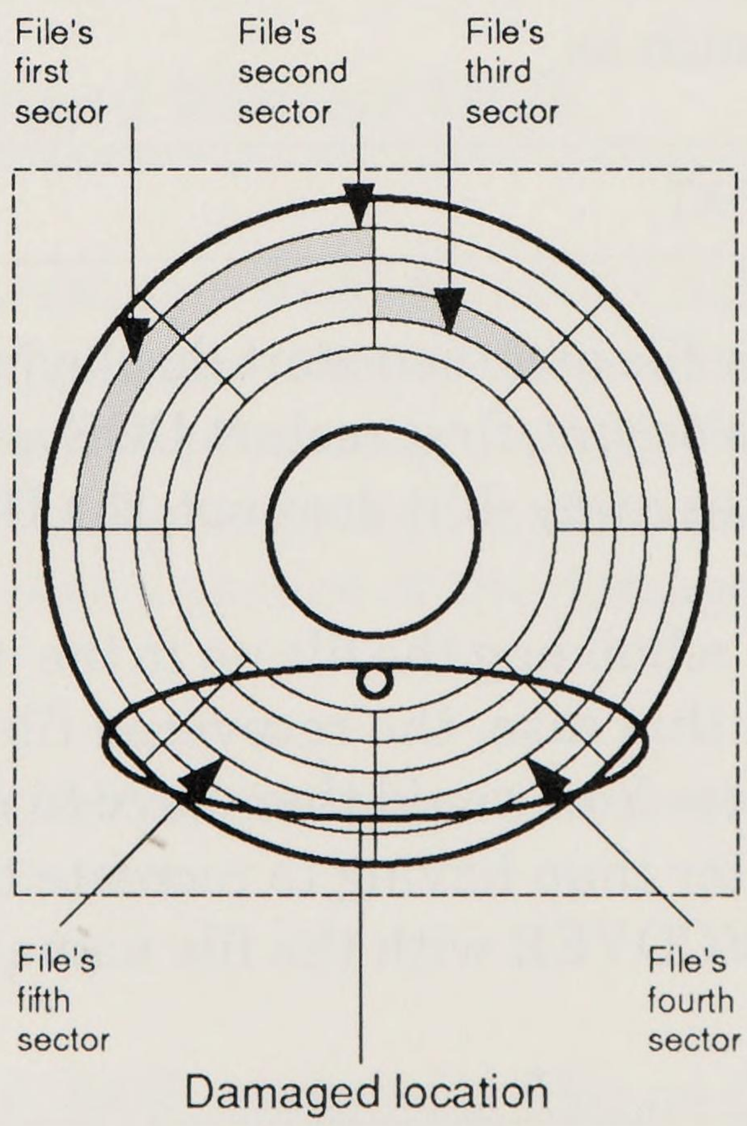


Figure 16-3. Lost sector on a disk

Trap: *Just as RECOVER can save a portion of a damaged disk, it can also destroy the information on a previously undamaged disk. Never use RECOVER on a file or disk that does not contain damaged sectors.*

RECOVER is a last resort. Later, this book will discuss the DOS BACKUP command and the importance of always maintaining duplicate copies of your files. In so doing, you should never require RECOVER. If one of your files becomes damaged, use the backup copy of the file.

If you have to recover a complete disk, use the command

```
A> RECOVER B:
```

RECOVER will begin rebuilding as many of the files on the disk as it can. Each time RECOVER successfully builds a file, it will create a file in the root directory of the disk with a name in the form

```
FILE0001.REC
```

For example:

```
Volume in drive A has no label
Directory of A:\

FILE0001      REC        14336   7-06-88  6:35a
FILE0002      REC         6144   7-06-88  6:35a
FILE0003      REC         2048   7-06-88  6:35a
FILE0004      REC        89088   7-06-88  6:35a
FILE0005      REC         3072   7-06-88  6:35a
FILE0006      REC         2048   7-06-88  6:35a
FILE0007      REC        63488   7-06-88  6:35a
FILE0008      REC        10854   7-06-88  6:35a
          8 File(s)    73728 bytes free
```

Once RECOVER builds these files, you can use the DOS TYPE command to determine the file's contents, renaming it appropriately.

```
A> TYPE FILE0001.REC
```

Trap: *Remember that DOS allows only a fixed number of files in the root directory. If RECOVER creates more files than DOS will allow, it will display an error and terminate.*

If this error message occurs, copy several of the recovered files to a new disk, deleting them from the original disk so that you can again invoke the RECOVER command for the remaining files. Keep in mind that RECOVER is a last resort.

Setting and Displaying File Attributes

Earlier in this book you learned that write protecting your disks can protect them from errant DOS commands such as

```
A> DEL *.*
```

However, if you are using a fixed disk, you cannot place a write-protect tab on your disk. Instead, DOS allows you to set the files that you do not want DOS to change or delete to read-only. A *read-only file* can be read by DOS for displaying, printing, or execution, but it cannot be modified or deleted.

Tip: *The DOS ATTRIB command allows you to set one or more files to read-only.*

Create the following file:

```
A> COPY CON B:NOTES.TXT
The DOS ATTRIB command allows
you to set a file to read only.
^Z
        1 File(s) copied
```

Using the DOS ATTRIB command

```
A> DIR  A:ATTRIB.EXE

Volume in drive A is MS330PP01
Directory of  A:\

ATTRIB          EXE          10656   7-24-87  12:00a
            1 File(s)      5120 bytes free
```

you can set the file to read-only, as shown here:

```
A> ATTRIB  +R  B:NOTES.TXT
```

When this command completes, use the DOS TYPE command to display the file's contents.

```
A> TYPE  B:NOTES.TXT
The DOS ATTRIB command allows
you to set a file to read only.

A>
```

Next, try to delete the file.

```
A> DEL  B:NOTES.TXT
Access denied

A>
```

A directory listing of drive B shows that the NOTES.TXT file still exists.

```
A> DIR  B:

Volume in drive B has no label
Directory of  B:\

NOTES           TXT      63        7-06-88  6:37a
            1 File(s)      361472 bytes free
```

Since you marked the file as read-only, DOS can read the file but cannot change or delete it.

This is what happens if you attempt to overwrite the contents of the file:

```
A> COPY  FILENAME.EXT  B:NOTES.TXT
File creation error
            0 File(s) copied

A>
```

As you can see, once you set a file to read-only, it is protected from many DOS commands.

Tip: *The DOS ATTRIB command supports DOS wildcard characters.*

To set all of the DOS commands in your DOS subdirectory, for example, use the command

```
C> ATTRIB  +R  \DOS\*.*
```

Tip: *Just as the +R qualifier directs ATTRIB to set a file to read-only, the -R qualifier directs ATTRIB to set the file back to read and write access, which allows DOS to modify or delete the file.*

To change the NOTES.TXT file back to DOS read and write access, use the command

```
A> ATTRIB  -R  B:NOTES.TXT
```

If you again attempt to delete the file

```
A> DEL  B:NOTES.TXT
```

DOS will delete the file as requested.

```
A> DIR  B:NOTES.TXT
File not found

A>
```

To simplify backup procedures (discussed later in this book), DOS assigns an *archive,* or backup, *required attribute* to your files. Each time you modify or create a DOS file, DOS sets the archive attribute. When set, this attribute tells DOS that the file has been recently created or changed and needs to be backed up. When the DOS BACKUP command later makes a copy of the file, BACKUP clears the archive required attribute. Later in this book you will learn how to direct BACKUP to back up only files that have been created or modified since the last BACKUP command (as opposed to all of the files on your disk). DOS does this by examining each file's *archive required bit* and selecting only those files with the attribute set. This saves you considerable time.

Tip: *The +A qualifier directs ATTRIB to set a file's archive required attribute. Likewise, the -A qualifier directs ATTRIB to clear this attribute.*

Chapter 17 will discuss the DOS XCOPY command, which allows you to perform selective file copies. At that time you will learn how to copy just those files that have their archive required attribute set. You should gain a better understanding of this qualifier during that discussion.

Tip: *If you invoke ATTRIB without specifying a qualifier (that is, the +R, -R, +A, or -A), ATTRIB will display the current attributes of the file(s) specified in the command line.*

With your Program Disk in drive A, copy the files with the COM extension to drive B.

```
A> COPY *.COM B:
```

Next, invoke the DOS ATTRIB command, as shown here.

```
A> ATTRIB  B:*.*
```

In this case, DOS will display

```
A          B:\NOTES.TXT
A          B:\ASSIGN.COM
A          B:\CHKDSK.COM
A          B:\COMMAND.COM
A          B:\COMP.COM
A          B:\DISKCOMP.COM
A          B:\DISKCOPY.COM
A          B:\EDLIN.COM
A          B:\FDISK.COM
A          B:\FORMAT.COM
A          B:\GRAFTABL.COM
A          B:\GRAPHICS.COM
A          B:\KEYB.COM
A          B:\LABEL.COM
A          B:\MODE.COM
A          B:\MORE.COM
A          B:\PRINT.COM
A          B:\RECOVER.COM
A          B:\SELECT.COM
A          B:\SYS.COM
```

Notice the uppercase "A" that precedes each file name. It stands for "archive required." Since you have just created the files on drive B, DOS has marked them as newly created—without a backup copy. If you set those files to read-only

```
A> ATTRIB  +R  B:*.*
```

and again display the file attributes

```
A> ATTRIB  B:*.*
```

DOS will display

```
A        R B:\NOTES.TXT
A        R B:\ASSIGN.COM
A        R B:\CHKDSK.COM
A        R B:\COMMAND.COM
A        R B:\COMP.COM
A        R B:\DISKCOMP.COM
A        R B:\DISKCOPY.COM
A        R B:\EDLIN.COM
A        R B:\FDISK.COM
A        R B:\FORMAT.COM
A        R B:\GRAFTABL.COM
A        R B:\GRAPHICS.COM
A        R B:\KEYB.COM
A        R B:\LABEL.COM
A        R B:\MODE.COM
A        R B:\MORE.COM
A        R B:\PRINT.COM
A        R B:\RECOVER.COM
A        R B:\SELECT.COM
A        R B:\SYS.COM
```

In this case, the uppercase "R" in the output stands for "read-only."

Tip: *The /S qualifier directs ATTRIB to apply the command to subdirectories that reside below the subdirectory specified in the command line.*

This command, for example, directs DOS to set all of the files on drive B to read-only, regardless of the subdirectories in which the files reside.

```
A> ATTRIB +R B:\*.* /S
```

Tip: *ATTRIB can be used to display all of the files on your disk.*

As just discussed, if you don't specify +R, -R, +A, or -A in your ATTRIB command, ATTRIB will display the current attributes for the specified files. If you use the /S qualifier, as shown here

```
C> ATTRIB \*.* /S
```

ATTRIB will display all of the files on your disk.

Comparing the Contents of Two Files

Although the DOS COMP command has existed since the advent of DOS to compare the contents of two files, it is most useful for ensuring that a file-copy operation was successful. The format of the COMP command is quite straightforward.

```
[drive:][path]COMP [primary_file] [secondary_file]
```

where the following is true:

drive specifies the disk drive containing the COMP.COM file. If you do not specify a disk drive, DOS uses the current default.

path specifies the subdirectory containing the COMP.COM file. If you do not specify a subdirectory, DOS uses the current default.

primary_file is the name of the first file that COMP is to compare.

secondary_file is the name of the second file that COMP is to compare.

As you can see by the command format, COMP is an external DOS command. If you omit the optional file names from your command line, COMP will prompt you for them, as shown here:

```
A> COMP

Enter primary file name NOTES.TXT

Enter 2nd file name of drive id
```

Assuming that the files COMP examines are identical, COMP will display the message

```
Files compare ok
```

If the files differ, COMP will display the byte location within the file that differs, as well as the values that differ.

Trap: *COMP displays the locations of only the first ten differences between two files.*

If the files that you are comparing contain many differences, COMP will display only the first ten differences, terminating the comparison and displaying the message

```
10 Mismatches - ending compare

Compare more files (Y/N)?
```

To better understand how COMP works, create the following files:

```
A> COPY CON B:VOWELS.UP
A
E
I
O
U
^Z
        1 File(s) copied

A> COPY CON B:VOWELS.LOW
a
e
i
o
u
^Z
        1 File(s) copied
```

The command

```
A> COMP B:VOWELS.UP B:VOWELS.LOW
```

will display

```
C:VOWELS.UP and C:VOWELS.LOW

Compare error at OFFSET 0
File 1 = 41

File 2 = 61

Compare error at OFFSET 3
File 1 = 45

File 2 = 65

Compare error at OFFSET 6
File 1 = 49

File 2 = 69

Compare error at OFFSET 9
File 1 = 4F

File 2 = 6F

Compare error at OFFSET C
File 1 = 55

File 2 = 75

Eof mark not found
```

The output of the COMP command can at first appear quite confusing. However, if you look at the chart in Appendix A, you will find that the hexadecimal value 41 corresponds to the letter "A," while the value 61 corresponds to the letter "a." As discussed, DOS uses the CTRL-Z character to represent the end of a text file. COMP always checks to see if CTRL-Z is the last character in the file. In the case of an EXE or COM (binary) file, DOS will likely not find CTRL-Z and will therefore display the message

```
Eof mark not found
```

In most cases, you can ignore this message.

For most users, COMP is best used as a safety check to ensure that your file copies are successful. For example, if you issue the command

```
A> COPY FILENAME.EXT B:
```

followed by the command

```
A> COMP FILENAME.EXT B:FILENAME.EXT
```

the COMP command should display the message

```
Files compare ok
```

as long as the file copy was successful. If COMP encounters differences, however, retry the COPY operation to a different target disk.

Tip: *COMP supports the DOS wildcard characters.*

If you issue the command

```
A> COMP *.* B:*.*
```

COMP will match files on both disks and compare them, displaying the result of each comparison on your screen.

Using FC to Compare Files

With MS-DOS version 3.3, the developers finally provided a file comparison utility: the FC command. FC (which stands for File Compare) allows you to compare two files, as shown here:

```
A> FC B:VOWELS.UP B:VOWELS.LOW
```

Tip: *FC supports several command-line qualifiers. If you invoke FC without specifying any command-line arguments, FC will display all of the qualifiers that it supports.*

FC displays all of the qualifiers, as shown here:

```
A> FC usage: fc [/a] [/b] [/c] [/l] [/lbNN] [/w] [/t] [/n] [/NNNN] file1 file2
```

The following list summarizes all of the qualifiers that FC supports

/A	Abbreviates the output of an ASCII (text) file comparison. FC only displays each set of lines that begin and end each set of differences as opposed to again the first displaying matching line.
/B	Directs FC to perform a binary comparison of the files.
/C	Directs FC to ignore the case of letters. Therefore, "A" and "a" would be considered equivalent.
/L	Directs FC to perform an ASCII (text) file comparison.
/Lb	Specifies the number of lines in the internal buffer that FC uses during its processing. By default, FC uses 100. If two files have more than this many differing lines, FC terminates the comparison.
/N	Displays the line numbers of the lines that differ during an ASCII file comparison.
/T	Directs FC not to treat a tab character as spaces. By default, FC considers a tab character as spaces equal to the next 8 character tab stop.
/W	Directs FC to consider successive tabs or space characters as one tab or space.
/nnnn	Specifies the number of lines that must match in succession before FC considers the file comparison resynched. By default, FC uses 2.

Create the following files.

```
A> COPY  CON  B:WEEK.1
Dodgers
Giants
Astros
Reds
Padres
Braves
^Z
            1 File(s) copied

A> COPY  CON  B:WEEK.2
Dodgers
Giants
Reds
Astros
Padres
Braves
^Z
            1 File(s) copied

A>
```

In this case, the command

```
A> FC  B:WEEK.1  B:WEEK.2
```

will display

```
***** B:WEEK.1
Giants
Astros
Reds
Padres
***** B:WEEK.2
Giants
Reds
Astros
Padres
*****
```

If you want to know the actual line numbers that differ, use the /N qualifier, as shown here:

```
***** B:WEEK.1
    2: Giants
    3: Astros
    4: Reds
    5: Padres
***** B:WEEK.2
    2: Giants
    3: Reds
    4: Astros
    5: Padres
*****
```

Notice that the effect of the /A qualifier is to abbreviate output in this comparison.

```
A> FC /A B:WEEK.1 B:WEEK.2
***** B:WEEK.1
Giants
...
Padres
***** B:WEEK.2
Giants
...
Padres
*****
```

Given the VOWELS.UP and VOWELS.LOW files that you created earlier in this chapter, the command

```
A> FC B:VOWELS.UP B:VOWELS.LOW
```

would display

```
***** B:VOWELS.UP
A
E
I
```

```
O
U
***** B:VOWELS.LOW
a
e
i
o
u
*****
```

However, if you instead include the /C qualifier, FC will ignore the case of the letters. As such, the command

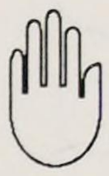
```
A> FC /C B:WEEK.1 B:WEEK.2
```

will display

```
fc: no differences encountered
```

Trap: *If FC encounters too many errors, it will terminate the file comparison, displaying the message*

```
resynch failed. Files are too different.
```



Trap: *FC does not support the DOS wildcard characters.*

If you specify a wildcard character in the FC command line, FC will display

```
fc: cannot open *.* - No such file or directory
```

Tip: *FC displays the differences between binary files differently than it does text-file differences.*

If you invoke FC to compare two binary (EXE or COM) files, FC displays differences in the form

```
aaaaaaaa byte1 byte2
```

where *aaaaaaaa* is the byte address within the file that differs, and *byte1* and *byte2* are the differing values. For example, the FC command

```
A> FC  TEST.EXE  OLDTEST.EXE
```

might display

```
00000000: 4D E9
00000001: 5A AE
00000002: 9A 39
00000003: 00 E9
00000004: 06 98
00000005: 00 39
00000006: 01 E9
00000007: 00 11
00000008: 20 28
00000009: 00 E9
0000000A: 00 B7
0000000B: 00 39
0000000C: FF 05
0000000D: FF 00
```

As you can see, FC is much more useful than the DOS COMP command.

Performing Selective File Replacements

If you are a fixed-disk user, you probably have several different software packages installed on your fixed disk, such as a word processor, a spreadsheet, and DOS. Each time a new version of these software packages is released, you must update your fixed disk accordingly. If you are using DOS version 3.2 or later, the REPLACE command allows you to simplify disk updates.

Tip: *REPLACE is an external DOS command that allows you to simplify disk updates.*

The format of the REPLACE command is

```
[drive:][path]REPLACE source_path target_path
[/A][/P][/R][/S][/W]
```

where the following is true:

drive specifies the disk drive containing the REPLACE.EXE file. If you omit the disk drive, DOS will use the current default.

path is the name of the subdirectory that contains the REPLACE.EXE file. If you omit the subdirectory name, DOS will use the current subdirectory.

source_path is the DOS path that contains the new files to be used in the disk update.

target_path is the DOS path into which the updated files are to be placed.

/A directs REPLACE to add only files to the target path that do not currently exist in that path.

/P directs REPLACE to prompt the user with

```
Replace FILENAME.EXT? (Y/N)
```

for each file, allowing you to perform a selective file replacement.

/R directs REPLACE to replace even files that are currently marked as read-only.

/S directs REPLACE to replace even files that reside in subdirectories below the target path.

/W directs REPLACE to prompt

```
Press any key to begin replacing files
```

which allows you to place the source floppy disk in the current drive, or the target disk in the target drive before the command executes.

Create the following files on drive B:

```
A> COPY  CON  B:ONE.DAT
ONE
^Z
            1 File(s) copied

A> COPY  CON  B:TWO.DAT
TWO
^Z
            1 File(s) copied

A> COPY  CON  B:THREE.DAT
THREE
^Z
            1 File(s) copied

A>
```

Next, remove that disk and insert a newly formatted disk in drive B. With your Program Disk in drive A, issue the command

```
A> REPLACE  *.*  B:  /W
```

When REPLACE prompts

```
Press any key to begin replacing file(s)
```

remove your Program Disk and insert the file that contains your newly created files. When you press ENTER, REPLACE will update the disk in drive B, as shown here:

```
Replacing B:\ONE.DAT

Replacing B:\TWO.DAT

Replacing B:\THREE.DAT
```

A directory listing of the disk in drive B reveals

```
A> DIR  B:

Volume in drive B has no label
Directory of  B:\

ONE          DAT        5        7-06-88  7:02a
TWO          DAT        5        7-06-88  7:02a
THREE        DAT        7        7-06-88  7:02a
          3 File(s) 358400 bytes free
```

Next, leaving the current disk in drive A, create the following files:

```
A> COPY  CON  FOUR.DAT
FOUR
^Z
          1 File(s) copied

A> COPY  CON  FIVE.DAT
FIVE
^Z
          1 File(s) copied

A>
```

Create a subdirectory called TEMP on drive B.

```
A> MKDIR  B:\TEMP
```

Copy the ONE.DAT file to this subdirectory.

```
A> COPY  ONE.DAT  B:\TEMP
          1 File(s) copied
```

Next, reinsert your Program Disk in drive A and again issue the REPLACE command. This time, however, include the /S qualifier.

```
A> REPLACE  *.*  B:  /W  /S
```

As before, when REPLACE prompts

```
Press any key to begin replacing file(s)
```

insert the disk containing the replacement files into drive A and press ENTER. REPLACE will again update the disk on drive A displaying

```
Replacing B:\TEMP\ONE.DAT

Replacing B:\ONE.DAT

Replacing B:\TWO.DAT

Replacing B:\THREE.DAT
```

Tip: *The /S qualifier directs REPLACE to find every occurrence of a replacement file on the target drive, and to replace it.*

In this case, REPLACE updated even the contents of the ONE.DAT file in the TEMP subdirectory.

Repeat this process, this time using the /A qualifier in place of /S.

```
A> REPLACE *.* B: /W /A
```

In this case, REPLACE will display

```
Adding B:\FOUR.DAT

Adding B:\FIVE.DAT
```

Tip: *The /A qualifier directs REPLACE to add only the replacement files to the target disk that do not currently reside there.*

Rather than replacing the contents of the files ONE.DAT, TWO.DAT, and THREE.DAT, REPLACE added the FOUR.DAT and FIVE.DAT files to the target disk. A directory listing of drive B reveals

```
A> DIR  B:

Volume in drive B has no label
Directory of  B:\

TEMP            <DIR>          7-06-88  7:06a
ONE       DAT        5         7-06-88  7:02a
TWO       DAT        5         7-06-88  7:02a
THREE     DAT        7         7-06-88  7:02a
FOUR      DAT        6         7-06-88  7:05a
FIVE      DAT        6         7-06-88  7:05a
        6 File(s)     355328 bytes free
```

Trap: *REPLACE will not update hidden files.*

You cannot use REPLACE to update to a new version of DOS, because REPLACE will not replace the contents of a hidden file.

Tip: *To enhance your batch processing capabilities, REPLACE provides you with the following exit status codes:*

Exit Code	Meaning
10	Successful
11	Error in the DOS command line
12	File not found
13	Path not found
15	Access denied
18	Insufficient memory available
15	Invalid disk drive specified

Tips and Traps Summary

Revisiting the FIND Command

Tip: The DOS FIND command allows you to include one or more file names in the command line. If present, FIND will examine each of the specified files one at a time for the given word or phrase.

Recovering a Damaged File or Disk

Tip: The DOS RECOVER command can sometimes save a portion of a damaged file or disk.

Trap: Just as RECOVER can save a portion of a damaged disk, it can also destroy the information on a previously undamaged disk. Never use RECOVER on a file or disk that does not contain damaged sectors.

Trap: Remember that DOS allows only a fixed number of files in the root directory. If RECOVER creates more files than DOS will allow, it will display an error and terminate.

Setting and Displaying File Attributes

Tip: The DOS ATTRIB command allows you to set one or more files to read-only.

Tip: The DOS ATTRIB command supports DOS wildcard characters.

Tip: Just as the +R qualifier directs ATTRIB to set a file to read-only, the -R qualifier directs ATTRIB to set the file back to read and write access, which allows DOS to modify or delete the file.

Tips and Traps Summary (*continued*)

Tip: The +A qualifier directs ATTRIB to set a file's archive required attribute. Likewise, the -A qualifier directs ATTRIB to clear this attribute.

Tip: If you invoke ATTRIB without specifying a qualifier (that is, the +R, -R, +A, or -A), ATTRIB will display the current attributes of the file(s) specified in the command line.

Tip: The /S qualifier directs ATTRIB to apply the command to subdirectories that reside below the subdirectory specified in the command line.

Tip: ATTRIB can be used to display all of the files on your disk.

Comparing the Contents of Two Files

Trap: COMP displays the locations of only the first ten differences between two files.

Tip: COMP supports the DOS wildcard characters.

Using FC to Compare Files

Tip: FC supports several command-line qualifiers. If you invoke FC without specifying any command-line arguments, FC will display all of the qualifiers that it supports.

Trap: If FC encounters too many errors, it will terminate the file comparison, displaying the message

```
resynch failed. Files are too different.
```

Trap: FC does not support the DOS wildcard characters.

Tip: FC displays the differences between binary files differently than it does text-file differences.

Tips and Traps Summary (*continued*)

Performing Selective File Replacements

Tip: REPLACE is an external DOS command that allows you to simplify disk updates.

Tip: The /S qualifier directs REPLACE to find every occurrence of a replacement file on the target drive, and to replace it.

Tip: The /A qualifier directs REPLACE to add only the replacement files to the target disk that do not currently reside there.

Trap: REPLACE will not update hidden files.

Tip: To enhance your batch processing capabilities, REPLACE provides you with the following exit status codes:

Exit Code	Meaning
0	Successful
1	Error in the DOS command line
2	File not found
3	Path not found
5	Access denied
8	Insufficient memory available
15	Invalid disk drive specified

17

COPY Versus XCOPY

If you are using DOS 3.2 or later, DOS provides you with two methods of performing file-copy operations, COPY and XCOPY. Throughout this book, you have used the DOS COPY command extensively to duplicate your files. In this chapter you will learn several additional capabilities of the DOS COPY command, as well as learning the full functionality of the XCOPY command. By the end of this chapter you will have a good feel for which applications are best suited for each.

Exploring the COPY Command in Detail

The most important point to stress about the DOS COPY command is that if the target file in the file-copy operation does not change from the source

```
A> COPY FILENAME.EXT B:FILENAME.EXT
```

you do not have to specify the target file name (as long as the target disk or subdirectory differs from the source):

```
A> COPY FILENAME.EXT B:
```

As such, the following commands are equivalent in function:

```
A> COPY FILENAME.EXT B:FILENAME.EXT <=> A> COPY FILENAME.EXT B:
A> COPY *.* B:\SUBDIR\*.*           <=> A> COPY *.* B:\SUBDIR
A> COPY B:FILENAME.EXT FILENAME.EXT <=> A> COPY B:FILENAME.EXT
```

Throughout this book, we have used the COPY command in its simplest form:

```
A> COPY FILENAME.EXT B:FILENAME.EXT
```

However, the complete format of the DOS COPY command is

```
COPY source_file [/A][/B] [+source_file [/A][/B]]
     [target_file][/A][/B][V]
```

Before examining each of the qualifiers in the COPY format, you need to know the function of the plus sign that appears in it.

Tip: *COPY allows you to combine multiple files into one file by placing a plus sign between the source files to combine.*

To better understand this processing, create the following files:

```
A> COPY CON PART.1
This is line 1
^Z
        1 File(s) copied
```

```
A> COPY  CON  PART.2
This is line 2
^Z
            1 File(s) copied

A> COPY  CON  PART.3
This is line 3
^Z
            1 File(s) copied

A>
```

To combine these files into one large file, use the command

```
A> COPY  PART.1+PART.2+PART.3  COMBINED.DAT
```

In this case, COPY will respond with

```
A> COPY  PART.1+PART.2+PART.3  COMBINED.DAT
PART.1
PART.2
PART.3
            1 File(s) copied
```

If you display the contents of the COMBINED.DAT file

```
A> TYPE  COMBINED.DAT
This is line 1
This is line 2
This is line 3

A>
```

you can see that COPY has combined the files into one file as desired.

Tip: *By using the + operator in the COPY command line, you can add information to an existing file from the keyboard.*

Assume that you want to add the line

```
This is the last line
```

to the file

```
A> TYPE COMBINED.DAT
This is line 1
This is line 2
This is line 3

A>
```

To do so, invoke COPY as

```
A> COPY COMBINED.DAT+CON COMBINED.DAT
```

COPY will display the device name CON, which is your prompt to enter additional information:

```
A> COPY COMBINED.DAT+CON COMBINED.DAT
COMBINED.DAT
CON
```

Type the desired information and press F6:

```
A> COPY COMBINED.DAT+CON COMBINED.DAT
COMBINED.DAT
CON
This is the last line
^Z
        1 File(s) copied

A>
```

When you display this file,

```
A> TYPE  COMBINED.DAT
This is line 1
This is line 2
This is line 3
This is the last line

A>
```

you can see that DOS has appended the line of text as desired.

With today's much improved disk technology, it is possible, although very rare, that your disk drive does not actually record the information on disk as DOS intended, due to a hardware error in the disk drive. Depending upon the error, DOS may not even be aware of the fact that this error has occurred. As a safety check, COPY provides the /V qualifier.

Tip: *The /V qualifier directs COPY to verify that the information is correctly recorded on disk—by reading and comparing it to the original data that DOS copied.*

To use the /V qualifier, place it at the end of your command line, as shown here:

```
A> COPY  FILENAME.EXT  B:  /V
```

COPY will read all of the information that it writes to the disk, comparing it to the original information, as shown in Figure 17-1. If your disk does not record the information correctly, DOS is now aware of the error and can notify you of the verification error.

Earlier, this book stated that batch files and the memos that you create on your computer contain only the letters and numbers that are common to most texts, and are called *text* or *ASCII files.* Programs (COM and EXE files) are *binary files.* As you know, DOS treats text and binary files differently. For example, DOS cannot display the contents of a binary file with the TYPE or PRINT commands.

Tip: *The /A qualifier tells COPY that a file is an ASCII file, whereas the /B qualifier tells COPY that the file is a binary file.*

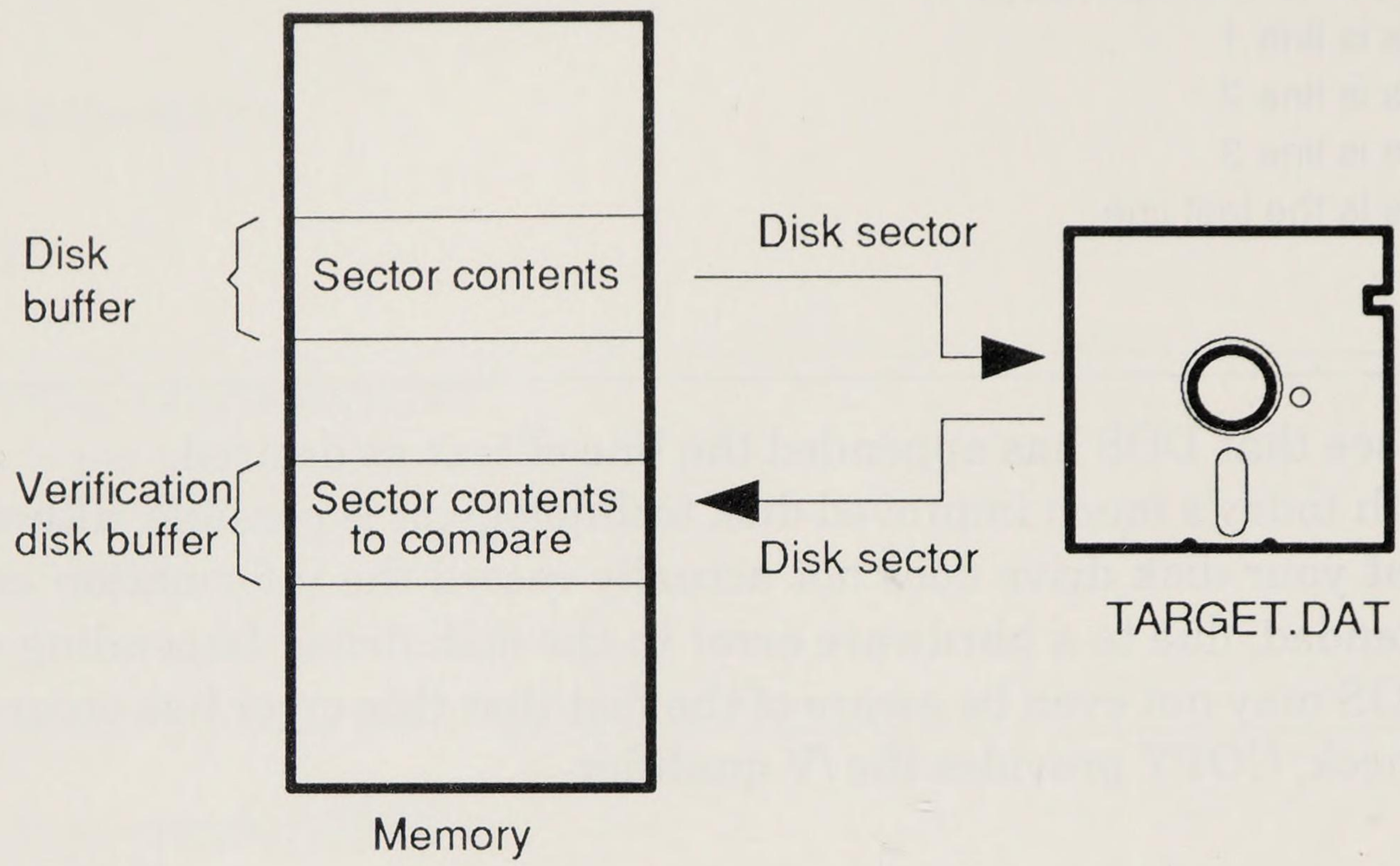


Figure 17-1. DOS verifies that target and source sectors are the same

By default, when DOS performs a standard file-copy operation

```
A> COPY FILENAME.EXT B:FILENAME.EXT
```

it assumes that both files are binary files. As it turns out, this assumption is perfectly valid for copying text or binary files. However, when you use COPY with the plus-sign operator to combine multiple files

```
A> COPY A+B C
```

DOS assumes that the files are text files. If this assumption is invalid, place the /B qualifier in the command line, as shown here:

```
A> COPY A/B+B/B C
```

DOS will perform a byte-by-byte copy of the file, ignoring possible embedded CTRL-Z (text end-of-file) characters.

Working with the DOS XCOPY Command

With your Supplemental Programs Disk in drive A, issue the command:

```
A> DIR  XCOPY.EXE

Volume in drive A is MS330PP02
Directory of  A:\

XCOPY          EXE          11216    7-24-87  12:00a
            1 File(s) 2048 bytes free
```

As you can see, XCOPY is an external DOS command that provides several extensions to the DOS COPY command. The format of the XCOPY command is

```
[drive:][path]XCOPY [/B] source_files [target_files] [/A] [/D:mm:dd:yy] [/E] [/M] [/S]
[/V] [/W]
```

where the following is true:

drive specifies the disk drive containing the XCOPY.EXE file. If you do not specify a disk drive identification, DOS uses the current default drive.

path specifies the subdirectory path containing the XCOPY.EXE file. If you do not specify a subdirectory path, DOS uses the current directory.

source_files specifies the file(s) to copy. XCOPY fully supports the DOS wildcard characters.

target_files specifies the name of the file(s) that XCOPY is to create. As with the COPY command, if the target file name is the same as the source file, you can omit the target file name.

/A directs XCOPY to copy only files whose archive required attribute is set (see the ATTRIB command in Chapter 16).

/B is used to create MCOPY.EXE under MS-DOS as discussed later in this chapter.

/D:mm:dd:yy directs XCOPY to copy only files created on or after the specified date.

/*E* directs XCOPY to create subdirectories on the target disk that match subdirectories on the source, even if the newly created directories will be empty.

/*M* directs XCOPY to copy only files whose archive required attribute is set (see the ATTRIB command in Chapter 16). When XCOPY successfully copies the file to the target disk, it will clear the file's archive required attribute.

/*P* directs XCOPY to prompt

```
FILENAME.EXT (Y/N)?
```

before copying each file to the target disk, allowing the user to perform a selective file-copy operation.

/*S* directs XCOPY to copy the subdirectories contained in the source (and the files that they contain) to the target disk.

/*V* directs XCOPY to verify that the data copied was successfully recorded on disk.

/*W* directs XCOPY to prompt

```
Press any key to begin copying file(s)
```

before beginning the file-copy operation, which allows the user to exchange floppy disks as required.

As you can see, XCOPY supports a myriad of command-line qualifiers. You have used several qualifiers very similar to these qualifiers earlier in this book with the ATTRIB and REPLACE commands. In general, XCOPY allows you to

- Perform "standard" file-copy operations such as

```
A> XCOPY FILENAME.EXT B:FILENAME.EXT
```

- Copy only those files created on or after the specified date:

```
A> XCOPY *.* B:*.* /D:12:08:88
```

- Copy only those files whose archive attribute is set.

```
A> XCOPY *.* B:*.* /A
```

- Copy the subdirectories that reside in the specified directory as well as the files that the subdirectories contain.

```
A> XCOPY *.* B:*.* /S
```

- Perform a verification check of the file copied to ensure the information was correctly recorded on disk:

```
A> XCOPY *.* B:*.* /V
```

To better understand XCOPY's capabilities, insert a formatted disk in drive A and create the following files. First, create the subdirectories ONE, TWO, and THREE, as shown here:

```
A> MKDIR \ONE
A> MKDIR \TWO
A> MKDIR \THREE
```

Second, create the following files:

```
A> COPY CON \ONE\1.DAT
1
^Z
            1 File(s) copied

A> COPY CON \TWO\2.DAT
2
^Z
            1 File(s) copied

A> COPY CON \ROOT.TXT
This file resides in the
root directory.
^Z
            1 File(s) copied
```

Next, place your Supplemental Programs Disk back in drive A and issue the command

```
A> XCOPY *.* B: /S /W
```

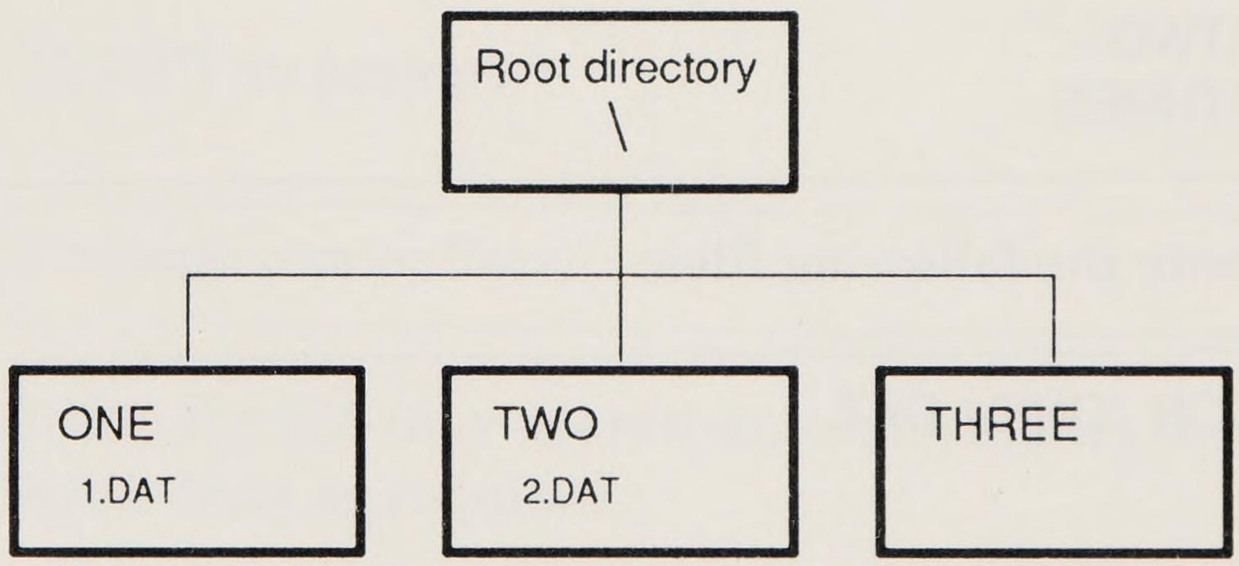
When XCOPY prompts

```
Press any key to begin copying file(s)
```

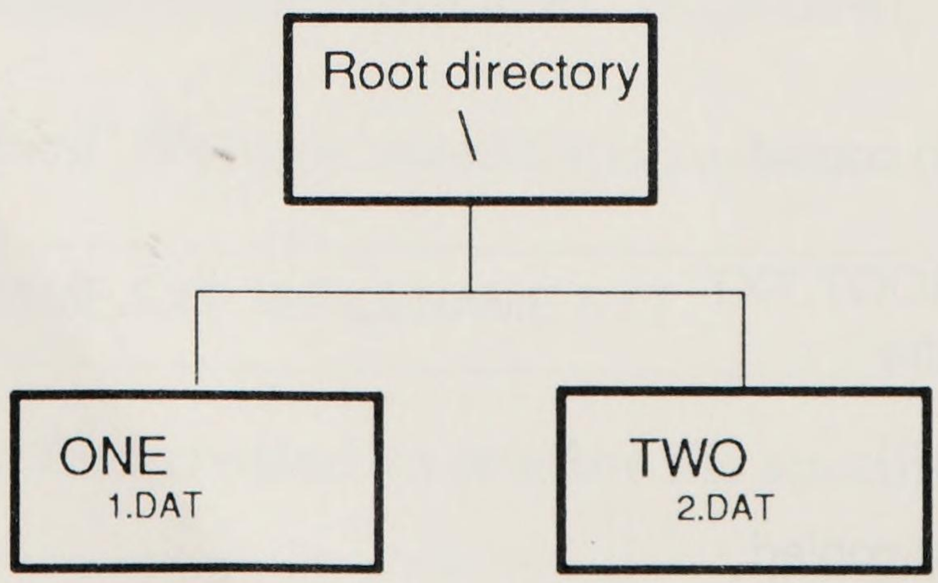
insert your disk containing the newly created files and press ENTER. XCOPY will respond with

```
Reading source file(s)...
A:ONE\1.DAT
A:TWO\2.DAT
            2 File(s) copied
```

In this case given the directory structure



XCOPY will create a target directory structure, which contains



Trap: *The /S qualifier directs XCOPY to copy only nonempty subdirectories to the target disk.*

In this case, since the THREE subdirectory does not contain any files, XCOPY did not create the subdirectory on the target disk.

Tip: *The /E qualifier directs XCOPY to create subdirectories on the target directory even if the directory does not contain files.*

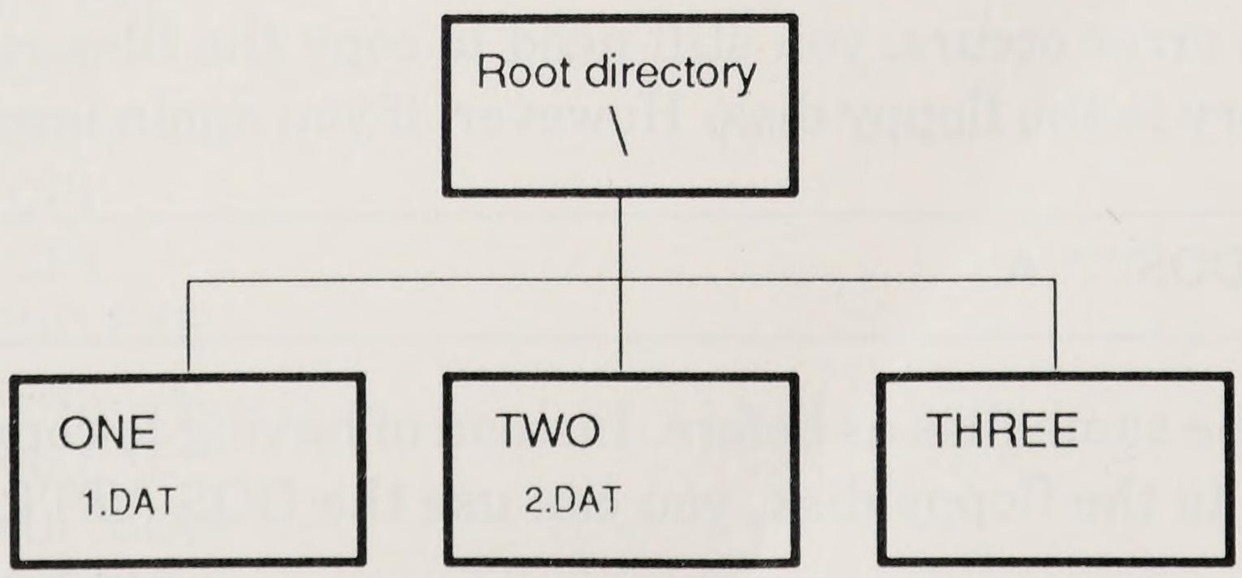
Repeat the previous XCOPY command, this time including the /E qualifier:

```
A> XCOPY *.* B: /S /W /E
```

Upon invocation, XCOPY will display

```
Press any key to begin copying file(s)

Reading source file(s)...
A:ONE\1.DAT
A:TWO\2.DAT
            2 File(s) copied
```

The /E qualifier, in this case, directed XCOPY to create an identical directory structure to the source disk. Although XCOPY does not display the fact that it created the empty directory THREE, the /E qualifier ensures that the directory is created, yielding the structure



Tip: *You can use XCOPY in conjunction with the DOS ATTRIB command to copy files from a fixed disk to a floppy disk.*

If you have tried to copy files from your fixed disk to a floppy disk, you may have encountered either the error

```
Insufficient disk space
```

or the error

```
File creation error
```

As previously discussed, the error occurred either because your floppy disk ran out of disk space or because you attempted to create more files in the disk's root directory than DOS will support. Unfortunately, once either of these errors occur, you have no easy way to continue the file-copy operation at the last file that DOS successfully copied.

For example, if you are using a fixed disk and have placed all of your Program Disk and Supplemental Programs Disk files into the subdirectory DOS, issue the command

```
C> COPY \DOS\*.* A:
```

to copy the files to a 360K floppy disk (remember, FORMAT /4 allows you to create a 360K disk in your 1.2MB disk drive). Once the floppy disk fills, DOS will display

```
Insufficient disk space
```

When this error occurs, you still need to copy the files remaining in the DOS subdirectory to the floppy disk. However, if you again issue the command

```
C> COPY \DOS\*.* A:
```

DOS will copy the same files as before. Instead of having to copy the remaining files one by one to the floppy disk, you can use the DOS ATTRIB and XCOPY commands.

First, using ATTRIB, set all of the files in the DOS subdirectory to archive required:

```
C> ATTRIB +A \DOS\*.*
```

Next, use the \M qualifier when you invoke the XCOPY command:

```
C> XCOPY \DOS\*.* A: /M
```

The /M qualifier directs XCOPY to copy only those files whose archive required attribute is set. Once XCOPY successfully copies a file, it clears the file's archive required attribute. In this case, XCOPY will display

```
Reading source file(s)...
\DOS\COMMAND.COM
\DOS\ANSI.SYS
\DOS\COUNTRY.SYS
\DOS\DISPLAY.SYS
\DOS\DRIVER.SYS
\DOS\FASTOPEN.EXE
\DOS\FDISK.COM
\DOS\FORMAT.COM
\DOS\KEYB.COM
\DOS\KEYBOARD.SYS
\DOS\MODE.COM
\DOS\NLSFUNC.EXE
\DOS\PRINTER.SYS
\DOS\REPLACE.EXE
\DOS\SELECT.COM
\DOS\SYS.COM
\DOS\VDISK.SYS
\DOS\XCOPY.EXE
\DOS\EGA.CPI
\DOS\LCD.CPI
\DOS\4201.CPI
\DOS\5202.CPI
\DOS\APPEND.EXE
\DOS\ASSIGN.COM
\DOS\ATTRIB.EXE
\DOS\BACKUP.COM
\DOS\BASIC.COM
```

```
\DOS\BASICA.COM
Insufficient disk space

        27 File(s) copied
```

When the disk in drive A fills, place a new floppy disk in drive A and resue the XCOPY command just shown. Since XCOPY has been clearing the archive required attribute of each file, it knows exactly at which file to continue copying:

```
Reading source file(s)...
\DOS\BASICA.COM
\DOS\CHKDSK.COM
\DOS\COMP.COM
\DOS\DEBUG.COM
\DOS\DISKCOMP.COM
\DOS\DISKCOPY.COM
\DOS\EDLIN.COM
\DOS\FIND.EXE
\DOS\GRAFTABL.COM
\DOS\GRAPHICS.COM
\DOS\JOIN.EXE
\DOS\LABEL.COM
\DOS\MORE.COM
\DOS\PRINT.COM
\DOS\RECOVER.COM
\DOS\RESTORE.COM
\DOS\SHARE.EXE
\DOS\SORT.EXE
\DOS\SUBST.EXE
\DOS\TREE.COM
\DOS\BASIC.PIF
\DOS\BASICA.PIF
\DOS\MORTGAGE.BAS
\DOS\EXE2BIN.EXE
\DOS\LIB.EXE
\DOS\LINK.EXE
\DOS\VDISK.ASM
        27 File(s) copied
```

Tip: *Under MS-DOS, the XCOPY command can be issued as either MCOPY or XCOPY. Depending on the method you use, the processing that XCOPY performs will differ.*

In some instances if XCOPY cannot determine whether the target for a file copy is a file or a DOS subdirectory, it will prompt you with

```
Does FILENAME specify a file name
or directory name on the target
(F = file, D = directory)? ^C
```

To eliminate this prompt, you can create the MCOPY command, which supports several assumptions to determine whether the target is a file name or subdirectory. Before you can invoke the DOS MCOPY command, you must create it using the command

```
A> XCOPY /B XCOPY.EXE MCOPY.EXE
```

MCOPY is very similar to XCOPY in function. However, MCOPY makes the following assumptions about the target:

- If the source file name is a DOS subdirectory name, MCOPY assumes that the target is also a subdirectory.
- If the source file name contains multiple files, MCOPY assumes that the target is a subdirectory.
- If the target file name ends with a backslash, MCOPY assumes that the target is a subdirectory.

Most users probably won't need this form of the XCOPY command. However, it is important to know that it exists in case you see the MCOPY.EXE file on another user's disk.

Tip: *To support batch processing, XCOPY provides the following exit status codes:*

Exit Code	Meaning
0	Successful file copy
1	No files found to copy
2	User terminated with CTRL-C
4	Initialization error
5	Critical error (drive open)

When to Use XCOPY Instead of COPY

Now that you have two means of performing file-copy operations, the question arises as to when to use each. The answer lies in what you want to accomplish. Since COPY is an internal DOS command, it is always readily available at the DOS prompt. Therefore, for "standard" file-copy operations, use COPY:

```
A> COPY FILENAME.EXT B:
```

Also, COPY allows you to combine multiple files:

```
A> COPY A+B+C D
```

If you attempt to combine files with XCOPY, DOS will display

```
A> XCOPY A+B+C D

Invalid parameter
          0 File(s) copied

A>
```

However, for batch processing, XCOPY provides exit status codes, which can be used in conjunction with the IF ERRORLEVEL command. Likewise, if you need to perform selective file copies, XCOPY provides the /P qualifier, which prompts the user with

```
FILENAME.EXT (Y/N)?
```

before each file copy, allowing individual files to be included or excluded. XCOPY also allows you to copy files based upon the file's creation date or archive status:

```
A> XCOPY *.* B: /A
```

Lastly, XCOPY allows you to copy the files that reside in subdirectories below the source location, creating an identical directory structure on the target disk:

```
A> XCOPY *.* B: /S /E
```

So, the decision of when to use XCOPY versus COPY is entirely dependent on the task you are trying to perform.

Tips and Traps Summary

Exploring the COPY Command in Detail

Tip: COPY allows you to combine multiple files into one file by placing a plus sign between the source files to combine.

Tip: By using the + operator in the COPY command line, you can add information to an existing file from the keyboard.

Tip: The /V qualifier directs COPY to verify that the information is correctly recorded on disk—by reading and comparing it to the original data that DOS copied.

Tip: The /A qualifier tells COPY that a file is an ASCII file, whereas the /B qualifier tells COPY that the file is a binary file.

Working with the DOS XCOPY Command

Trap: The /S qualifier directs XCOPY to copy only nonempty subdirectories to the target disk.

Tip: The /E qualifier directs XCOPY to create subdirectories on the target directory even if the directory does not contain files.

Tip: You can use XCOPY in conjunction with the DOS ATTRIB command to copy files from a fixed disk to a floppy disk.

Tip: Under MS-DOS, the XCOPY command xcan be issued as either MCOPY or XCOPY. Depending on the method you use, the processing that XCOPY performs will differ.

Tips and Traps Summary (*continued*)

Tip: To support batch processing, XCOPY provides the following exit status codes:

Exit Code	Meaning
0	Successful file copy
1	No files found to copy
2	User terminated with CTRL-C
4	Initialization error
5	Critical error (drive open)

18

Getting the Most from DOS System Commands

Just as Chapter 16 examined the DOS file-manipulation commands, this chapter discusses several DOS commands that affect how DOS behaves. Therefore, these commands are grouped under the category of system commands. Although you probably will not use each of these commands on a daily basis, it is important for you to know that they exist, and how each affects your system.

Verifying Your Disk-Output Operations

In Chapter 17 you learned that although it is rare, it is still possible for your disk to incorrectly record the information that DOS directs it to write to your disk. At that time, you found that the /V qualifier directs COPY and XCOPY to perform disk-verification operations in which they reread the data DOS has recorded on disk to compare it to the original data (see Figure 18-1).

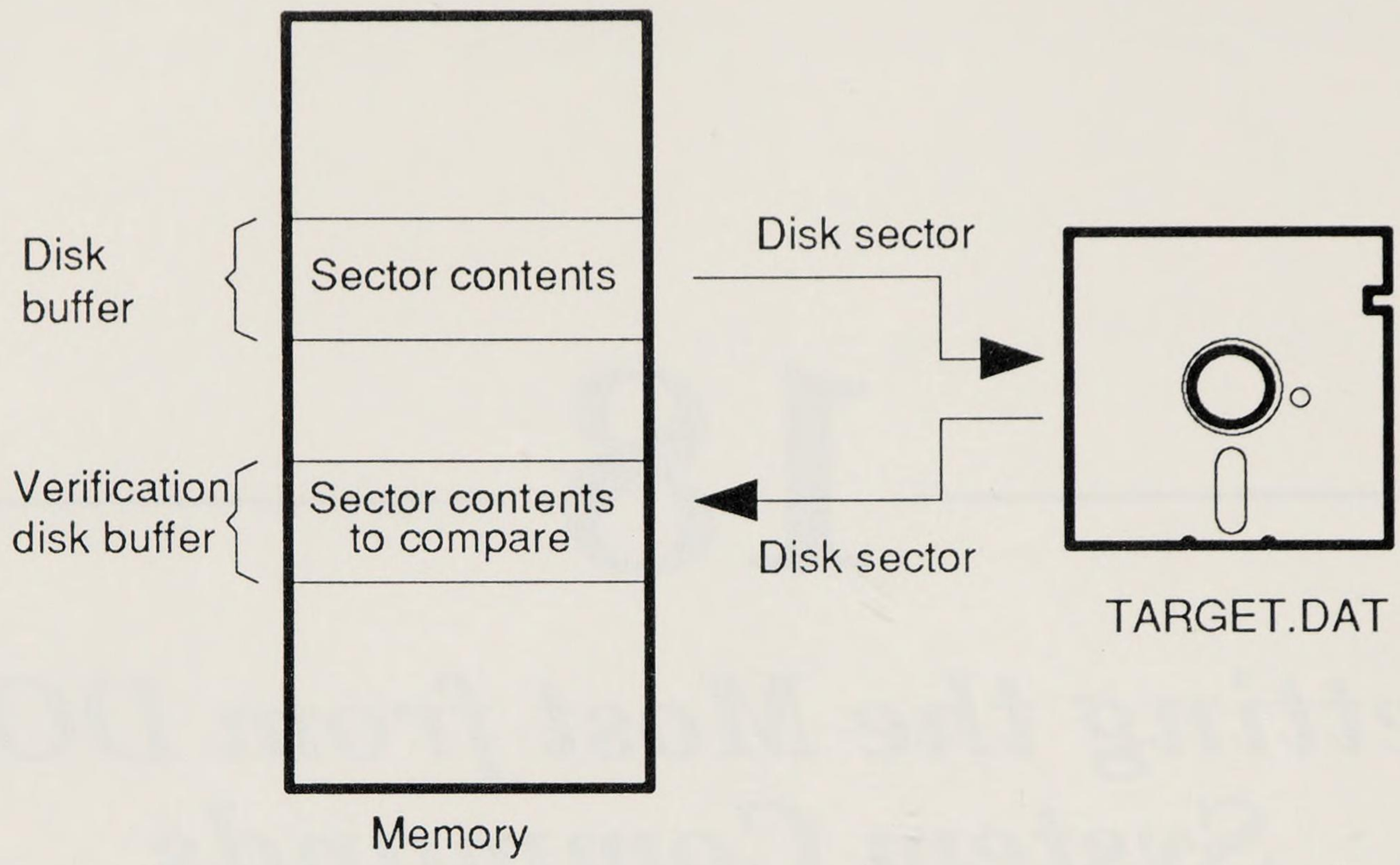


Figure 18-1. DOS verifies that target and source sectors are the same

However, most DOS commands do not provide a /V qualifier to invoke disk verification. Instead, in cases where you must ensure that your data is correctly recorded to disk, you can use the DOS VERIFY command.

Tip: *The DOS VERIFY command directs DOS to enable or disable disk verification for all disk output operations.*

To invoke disk verification, use the command

```
A> VERIFY ON
```

VERIFY is an internal DOS command. When you use it to enable disk verification as shown here, DOS will reread each sector that it writes to disk, ensuring that the data is correctly recorded.

Tip: *If you invoke VERIFY without specifying command-line parameters, VERIFY will display its current state (ON or OFF).*

At the DOS prompt, enter the command

```
A> VERIFY
VERIFY is on
```

As you can see, DOS displayed the current disk-verification state.

Trap: *Enabling disk verification for all DOS operations introduces system overhead, which decreases your overall system performance.*

If you use VERIFY to invoke disk verification, DOS will take about twice as long to perform your disk output operations, since it must now write the data, reread it from disk, and then compare the data to the original information. Since disk recording errors of this type are so rare, most users keep the default setting of

```
A> VERIFY OFF
```

in order to reduce their system overhead.

Changing the DOS Prompt

Throughout this book, the only time that your prompt has changed from

```
A>
```

is when you change default disk drives, as shown here:

```
A> B:
B>
```

The DOS PROMPT command, however, allows you to customize the system prompt to your liking.

Tip: *The DOS PROMPT command is an internal DOS command that directs DOS to set the system prompt to the specified series of characters.*

To set your prompt to the character string WHAT_NOW?, for example, enter the command.

```
A> PROMPT WHAT_NOW?
WHAT_NOW?
```

As you can see, DOS has changed your system prompt as you specified. If you press the ENTER key several times

```
A> PROMPT WHAT_NOW?
WHAT_NOW?
WHAT_NOW?
WHAT_NOW?
```

you can see that your prompt remains the same.

Trap: *PROMPT places an entry in the DOS environment that tells DOS the current prompt. If insufficient environment space exists, PROMPT will display the error message*

```
Out of environment space
```

and your prompt will not change. Issue the DOS SET command, which displays the environment entries.

```
WHAT_NOW? SET
COMSPEC=C:\COMMAND.COM
PATH=C:\DOS;C:\UTIL
PROMPT=WHAT_NOW?

WHAT_NOW?
```

As you can see, PROMPT has created the PROMPT= environment entry.

Tip: *To restore the DOS prompt to the system default (A>), type the PROMPT command without specifying any command-line parameters.*

At your current prompt, enter the DOS PROMPT command, as shown here:

```
WHAT_NOW? PROMPT
A>
```

As you can see, DOS resets your prompt to the system default. Likewise, if you issue the DOS SET command, you will find that DOS has removed the PROMPT= entry from the environment:

```
A> SET
COMSPEC=C:\COMMAND.COM
PATH=C:\DOS;C:\UTIL

A>
```

Tip: *To enhance your processing capabilities, PROMPT supports the following metacharacters:*

$b	The | character
$d	The current system date

$e	The ASCII escape character
$g	The greater-than character (>)
$h	The backspace character
$l	The less-than character (<)
$n	The default disk drive
$p	The current disk drive and directory
$q	The equal-sign character (=)
$t	The current system time
$v	The current DOS version number display
$$	The dollar-sign character
$_	The carriage return and linefeed characters

A *metacharacter* is a character that is preceded by a dollar sign, which has a predefined meaning. For example, to set your prompt to DOS>, use the command.

```
A> PROMPT DOS$g
DOS>
```

In this case, the metacharacter $g directs DOS to display the greater-than character. If you had instead attempted to specify the prompt as

```
A> PROMPT DOS>
```

DOS would assume that you were trying to perform I/O redirection and would display the error message.

```
File creation error
```

Using the $t metacharacter, this command sets your prompt to the current system time:

```
A> PROMPT $t
13:49:56.11
13:49:57.32
```

Tip: *If you are using DOS subdirectories, set your prompt to [$p]. In so doing, you cause DOS to always display your current directory on the screen display.*

Watch how the prompt [$p] changes as the current subdirectory changes:

```
C> PROMPT [$p]
[C:\] CD DOS
[C:\DOS] CD \WORDSTAR\LETTERS
[C:\WORDSTAR\LETTERS] B:
[B:\]
```

Note the use of the left and right brackets in the [$p]. By placing the brackets around the prompt, DOS groups the directory name within the brackets, as shown.

By using this prompt, your subdirectory traversal should become easier to follow.

Tip: *You can use the DOS PROMPT command in conjunction with the ANSI.SYS device driver.*

Earlier in this book you learned how to create files with EDLIN that use the ANSI driver to set your screen's background color to blue, or define the function keys on your keyboard. The DOS PROMPT command provides the $e metacharacter, which directs DOS to write the escape character. Therefore, if you have installed the ANSI.SYS device driver as discussed in Chapter 9, you can change your screen's background color to blue by using the command

```
A> PROMPT $e[44m
```

Once DOS sets your screen color, invoke PROMPT without any parameters, or with the desired system prompt. The background color—blue in this case—will remain in effect.

Table 18-1 illustrates the use of PROMPT to interface with each of the ANSI escape sequences. Remember that ANSI.SYS must be installed for these sequences to work.

Table 18-1. ANSI Escape Sequences Used with PROMPT

Sequence	Function	Example
Esc[Row;ColumnH	Set cursor at row, column	PROMPT $e[10;1H
Esc[*n*A	Move cursor up *n* rows	PROMPT $e[4A
Esc[*n*B	Move cursor down *n* rows	PROMPT $e[4B
Esc[*n*C	Move cursor right *n* columns	PROMPT $e[10C
Esc[*n*D	Move cursor left *n* columns	PROMPT $e[10D
Esc[6n	Get device status report	PROMPT $e[6n
Esc[s	Save cursor position	PROMPT $e[s
Esc[u	Restore cursor position	PROMPT $e[u
Esc[2J	Clear screen display	PROMPT $e[2J
Esc[nm	Set color attribute	PROMPT $e[44m

Note that you can combine several escape sequences into one PROMPT command. This command changes the system prompt to RED_BLUE> and displays the words in the appropriate colors:

```
A> PROMPT $e[41mRED_$e[44mBLUE$e[0m$g
```

Remember that you can also use the extended characters in your prompt. This example will change the prompt to the happy face followed by the greater-than symbol. To do so, enter **PROMPT**, followed by ALT-2, followed by the $g metacharacter. Upon invocation, prompt will change your prompt to

```
☻
```

As you can see, PROMPT has considerable functionality. Most users, however, should place the command

```
PROMPT [$p]
```

in their AUTOEXEC.BAT files ensuring that PROMPT always displays the current directory.

What Is CTTY?

In the past, some programmers connected a terminal (which is sometimes called a *TTY device*) to the serial port on their computers (see Figure 18-2). In such cases, the DOS CTTY command allowed them to enter commands from the terminal's keyboard, viewing the result of the command on the terminal's screen display. To use the terminal in place of the computer's screen and keyboard, the user entered the command

```
A> CTTY AUX
```

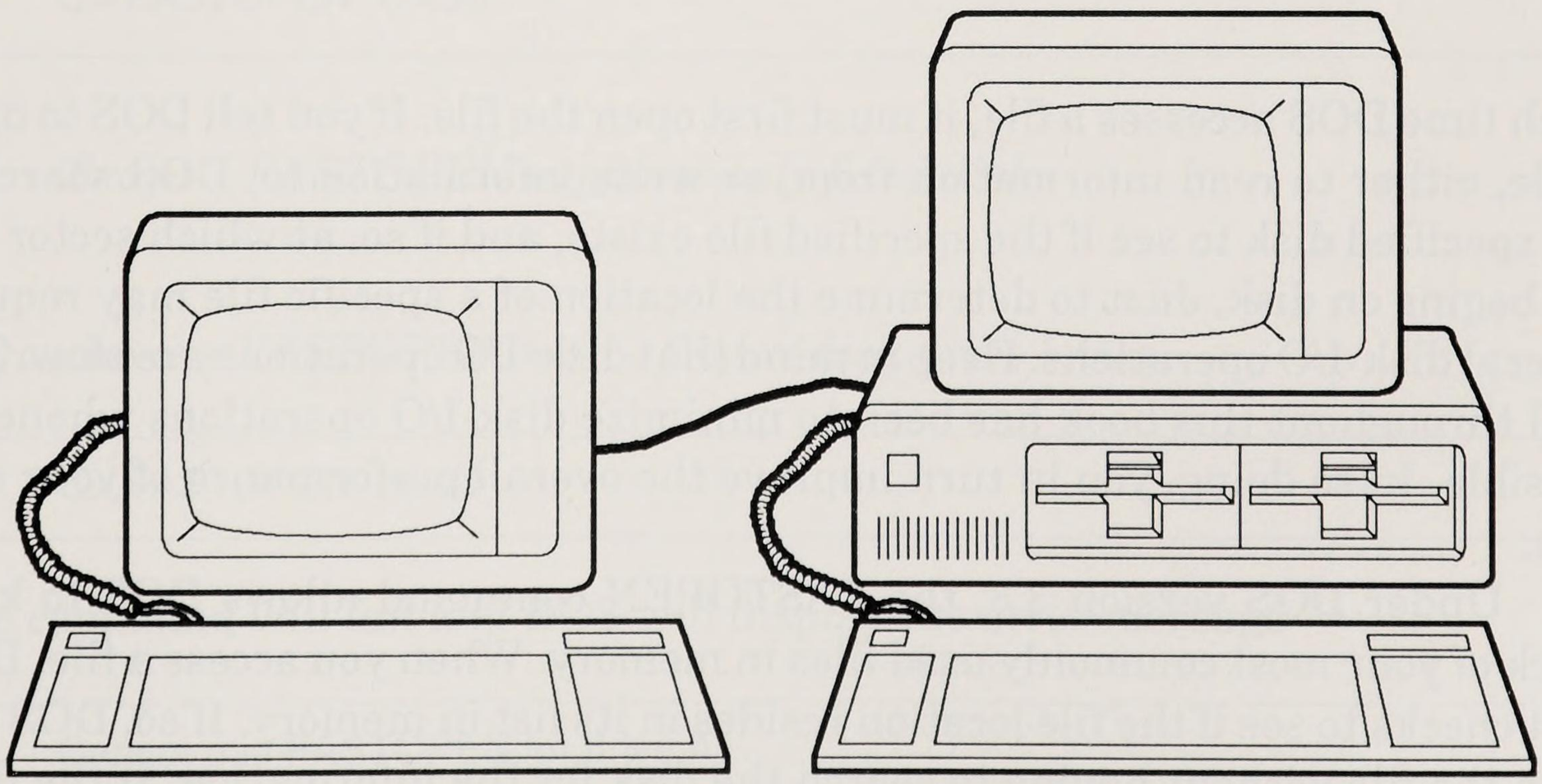


Figure 18-2. Connecting a terminal to the serial port

which directed DOS to obtain its input from the device connected to the *AUX device* (the first serial port). When the user later wanted to use the computer's keyboard and screen, he or she had to enter **CTTY CON** from the terminal's keyboard.

Tip: *Only use the DOS CTTY command when you have a terminal (TTY device) connected to your computer's serial port. If you invoke the command*

```
A> CTTY AUX
```

without such a device, you will have to reboot your system to continue.

Improving Disk-I/O Performance with FASTOPEN

Tip: *If you are using a fixed disk under DOS version 3.3, the FAST-OPEN command may improve your system performance by reducing the number of slow disk-I/O operations that DOS must perform.*

Each time DOS accesses a file, it must first open the file. If you tell DOS to open a file, either to read information from, or write information to, DOS searches the specified disk to see if the specified file exists, and if so, at which sector the file begins on disk. Just to determine the location of a specific file may require several disk-I/O operations. Keep in mind that disk-I/O operations are slow. The goal throughout this book has been to minimize disk-I/O operations whenever possible. In so doing, you in turn improve the overall performance of your system.

Under DOS version 3.3, the FASTOPEN command allows DOS to keep track of your most commonly used files in memory. When you access a file, DOS first checks to see if the file location resides in its list in memory. If so, DOS can open the file without having to search the disk for the information. If the file's information is not currently in memory, DOS will read the disk for the informa-

tion, and then update the list accordingly. Therefore, if you have several files that you continually use, DOS will always have their location on disk readily available, which in many cases will save you considerable time.

The format of the DOS FASTOPEN command is

```
[drive:][path]FASTOPEN fixed_disk:[=list_size][...]
```

where the following is true:

drive specifies the disk drive containing the FASTOPEN.EXE file. If you do not specify a disk drive, DOS uses the current default.

path specifies the subdirectory containing the FASTOPEN.EXE file. If you do not specify a subdirectory, DOS uses the current directory.

fixed_disk specifies the disk drive letter of the fixed disk, to keep track of files in memory.

list_size specifies the number of files in the list, from 10 to 999. The default size is 10.

... states that you can specify several disk drives in one command line.

This command, for example, allows DOS to track 30 files for drive C:

```
C> FASTOPEN  C:=30
```

Trap: *FASTOPEN supports only fixed disks.*

If you invoke FASTOPEN with a floppy-disk drive letter,

```
C> FASTOPEN  A:=30
```

the command will fail and DOS will display the error message

```
Cannot use FASTOPEN for drive A:
```

Trap: *FASTOPEN installs memory-resident software. As such, you can invoke FASTOPEN only once per computer session. To change FAST-OPEN parameters or specify additional disk drives, you must restart DOS.*

If you invoke FASTOPEN a second time, attempting to change settings, FAST-OPEN will display

```
FASTOPEN already installed
```

To change your FASTOPEN settings, you must restart DOS.

Trap: *Setting the number of FASTOPEN entries too large will slow down system performance.*

Keep in mind that once you install FASTOPEN, each time DOS must open a file, it will search the list of files that FASTOPEN is maintaining. If this list is long, DOS could spend considerable time simply searching the list. In addition, each FASTOPEN entry consumes 40 bytes of memory. If you use 999 entries, for example, you have consumed almost 40K of memory that is otherwise available to DOS and your applications.

Most users should get by with a FASTOPEN setting of about 30 files:

```
C> FASTOPEN  C:=30
```

The goal of FASTOPEN is to decrease the access time associated with your most commonly used files. If you do not continually use the same files, FAST-OPEN will not save you time.

Using the DOS EXIT Command

Many of your application programs (such as your word processor or spreadsheet) will allow you to leave the program temporarily to issue DOS commands. Once

you have completed the DOS commands, you can return to your program and continue right where you left off. Just how each application allows you to access DOS is specific to the application. However, the end result is that you obtain the DOS prompt:

```
A>
```

Once you complete your desired DOS commands, you can return to your application program by typing **EXIT**:

```
A> EXIT
```

When DOS acknowledges the EXIT command, it returns you to your application program.

Earlier in this book you learned how to invoke a nested batch file using COMMAND /C:

```
DATE
COMMAND /C BATFILE
TIME
```

In some cases, however, you may want the user to be able to stop the batch-file processing temporarily to execute other DOS commands. As you will see, by using COMMAND in conjunction with EXIT, you can do just that. With a disk that contains the COMMAND.COM file

```
A> DIR  COMMAND.COM

Volume in drive A is MS330PP01
Directory of  A:\

COMMAND      COM          25276   7-24-87  12:00a
            1 File(s)     5120 bytes free
```

create the batch file shown here:

```
A> COPY  CON  TEST.BAT
DATE
COMMAND
```

```
TIME
^Z
            1 File(s) copied
```

The DATE and TIME commands are self-explanatory. However, the line

```
COMMAND
```

needs additional discussion. When DOS encounters **COMMAND** without **/C command_name**, it will execute COMMAND. As discussed, COMMAND.COM contains your DOS internal commands and is responsible for displaying the DOS prompt. Therefore, when you invoke COMMAND, DOS displays a prompt from which you can issue subsequent DOS commands. When you are done entering DOS commands, type **EXIT** and DOS will resume processing the batch file at the TIME command.

In this case, the processing becomes

```
A> TEST

A> DATE
Current date is Sat 7-09-1988
Enter new date (mm-dd-yy):

A> COMMAND

The IBM Personal Computer DOS
Version 3.30 (C) Copyright International Business Machines Corp 1981, 1987
            (C) Copyright Microsoft Corp 1981, 1986

A> EXIT  }                    DOS prompt
                              available to user
A> TIME
Current time is 14:04:30.99
Enter new time:

A>
```

Tip: *The DOS EXIT command enables you to terminate a secondary command processor.*

Using DISKCOMP to Compare Two Floppy Disks

Just as the DOS DISKCOPY command allows you to copy the contents of one floppy disk to another, the DOS DISKCOMP command allows you to compare two disks. The format of the DISKCOMP command is

```
[drive:][path]DISKCOMP [first_disk] [second_disk] [/1][/8]
```

where the following is true:

drive specifies the disk drive containing the DISKCOMP.COM file. If you do not specify a disk drive letter, DOS will use the current default.

path specifies the DOS subdirectory that contains the DISKCOMP.COM file. If you do not specify a subdirectory, DOS will use the current directory.

first_disk is the disk drive letter that contains the first disk to compare. If you do not specify a disk drive, DOS will use the current default.

second_disk is the disk drive letter that contains the second disk to compare. If you do not specify a disk drive, DOS will use the current default.

/1 directs DISKCOMP to compare only one side of the disk.

/8 directs DISKCOMP to compare disks, assuming that each disk contains eight sectors per track.

Tip: *The DOS DISKCOMP command allows you to compare the contents of two floppy disks.*

This command, for example, directs DISKCOMP to compare the contents of the floppy disk in drive A, to the disk in drive B:

```
A> DISKCOMP  A:  B:
```

Upon invocation, DISKCOMP will display

```
Insert FIRST diskette in drive A:

Insert SECOND diskette in drive B:

Press any key when ready . . .
```

If you have a single-drive system, the command

```
A> DISKCOMP
```

directs DISKCOMP to prompt you for each disk during the compare operation, as shown here:

```
Insert FIRST diskette in drive A:

Press any key when ready . . .
```

If the disks compared are identical, DISKCOMP will display the message

```
Compare OK
```

However, if differences exist, DISKCOMP will display the differences in the form

```
Compare error on side 1, track 0
```

Trap: *DISKCOMP performs a sector-by-sector comparison of two disks. Just because two disks contain the same files does not mean that DISKCOMP will find the disks identical.*

As discussed, DOS stores your files on disk by placing the information that they contain into sectors. Just because two disks contain the same files does not mean that the files reside in the same sectors.

Trap: *DISKCOMP supports only floppy disks.*

If you invoke DISKCOMP with a fixed disk, DISKCOMP will display

```
Invalid drive specification
Specified drive does not exist,
or is non-removable
```

Trap: *DISKCOMP compares only disks of identical type.*

If, for example, you attempt to compare a 1.2MB disk to a 360K disk, DOS will display

```
Drive types or diskette types
not compatible

Compare process ended
```

Tip: *To support your batch processing, DISKCOMP provides the following exit status values:*

Exit Status	Meaning
0	Disks compare identically
1	Disks are different
2	User-terminated with CTRL-C
3	Read or write error during comparison
4	Initialization error

Enhancing Character Display with GRAFTABL

The IBM PC and PC compatibles periodically have difficulty displaying characters in the extended character set when the computer is in medium-resolution graphics mode. Although this problem does not affect most users, if you are run-

ning unique graphics applications, you may need to invoke the DOS GRAFTABL command.

Tip: *The DOS GRAFTABL command installs memory-resident software that helps the PC display extended characters in medium-resolution graphics mode.*

The format of the GRAFTABL command is

```
[drive:][path]GRAFTABL [codepage]
```

or

```
[drive:][path]GRAFTABL /STATUS
```

Chapter 19 will discuss codepages in detail. Essentially, a *codepage* is a character set. Valid codepages include the following:

Codepage	Country
437	U.S.
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

This command, for example, invokes GRAFTABL with the default (U.S.) codepage:

```
A> GRAFTABL
```

Upon invocation, GRAFTABL will display

```
No version of Graphic Character Set Table is already loaded.

USA version of Graphic Character Set Table has just been loaded.
```

Trap: *GRAFTABL loads memory-resident software. Therefore, you can invoke GRAFTABL only once per user session.*

If you attempt to invoke GRAFTABL a second time, DOS will display

```
USA version of Graphic Character Set Table is already loaded.
```

Tip: *To determine whether GRAFTABL has been loaded, and if so, with which character set, use the /STATUS qualifier to the GRAFTABL command.*

To determine the current status of the GRAFTABL command, use the command

```
A> GRAFTABL /STA
```

Most users will never require the GRAFTABL command.

File Sharing in a Local-Area Network

Over the past few years, more and more personal computers are being interconnected to share files and printers, as shown in Figure 18-3.

The DOS SHARE command provides file sharing in a local-area network. The format of the SHARE command is

```
[drive:][path]SHARE [/F:buffersize][/L:locks]
```

where the following is true:

drive is the disk drive containing the SHARE.EXE file. If you do not specify a disk drive letter, SHARE will use the current default.

path is the DOS subdirectory containing the SHARE.EXE file. If you do not specify a disk drive letter, SHARE will use the current subdirectory.

/F:buffersize specifies the number of bytes that SHARE will set aside for file-sharing information. The default size is 2048.

/L:locks specifies the number of file and record locks that SHARE will support. The default is 20.

Tip: *If your computer is not part of a local-area network, do not issue the DOS SHARE command.*

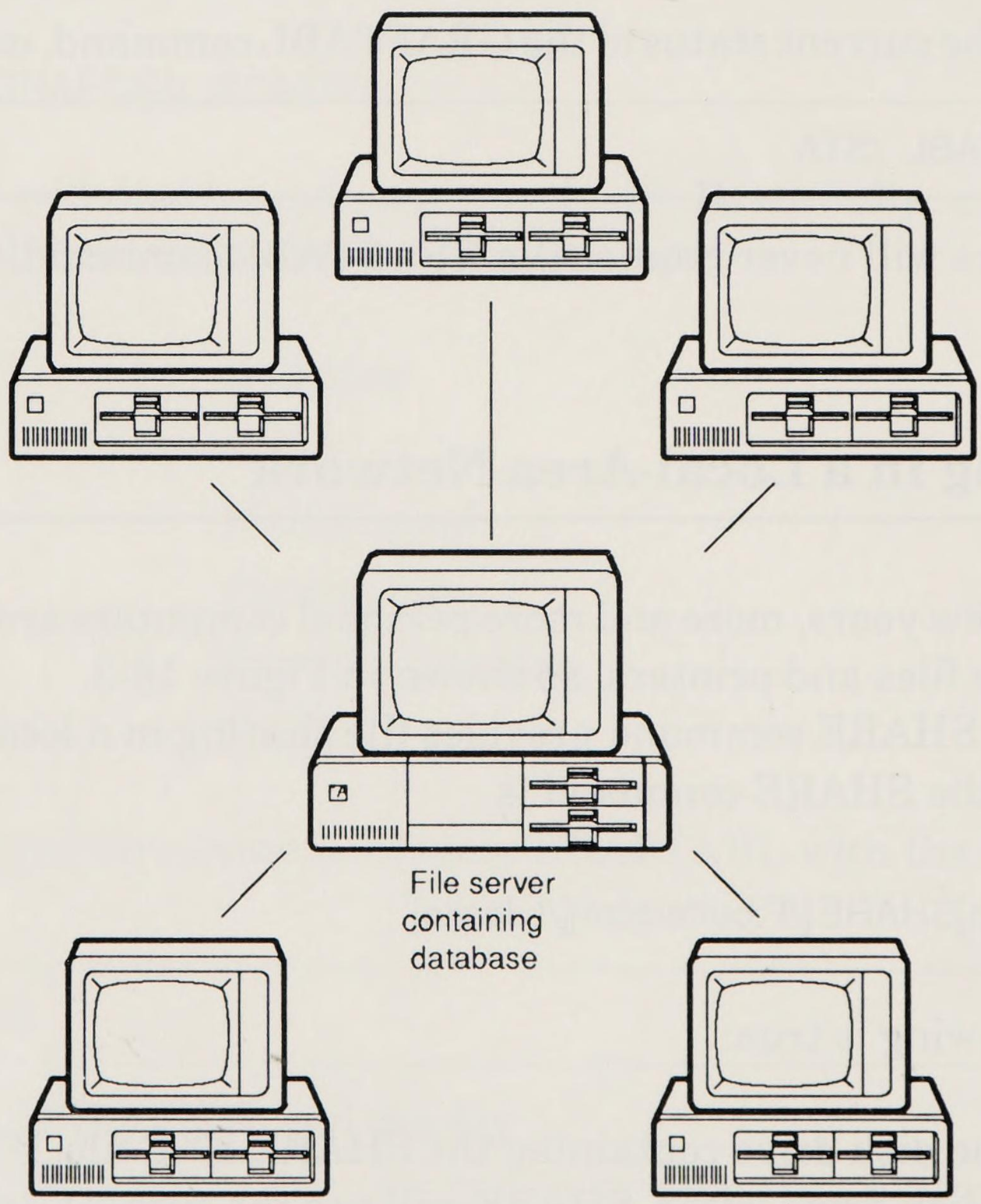


Figure 18-3. A local-area network

If you invoke SHARE when your computer is not part of a local-area network, DOS will needlessly check for file-sharing conflicts with each disk-I/O operation. These checks produce system overhead which, in turn, decreases your system performance.

In addition, the file-sharing buffer and locks consume memory that would otherwise be available to DOS.

Tip: *If your computer is part of a local-area network, have the individual who installed the network determine whether the SHARE command is required, and if so, with which settings.*

Tips and Traps Summary

Verifying Your Disk-Output Operations

Tip: The DOS VERIFY command directs DOS to enable or disable disk verification for all disk output operations.

Tip: If you invoke VERIFY without specifying command-line parameters, VERIFY will display its current state (ON or OFF).

Trap: Enabling disk verification for all DOS operations introduces system overhead, which decreases your overall system performance.

Changing the DOS Prompt

Tip: The DOS PROMPT command is an internal DOS command that directs DOS to set the system prompt to the specified series of characters.

Trap: PROMPT places an entry in the DOS environment that tells DOS the current prompt. If insufficient environment space exists, PROMPT will display the error message

```
Out of environment space
```

and your prompt will not change. Issue the DOS SET command, which displays the environment entries.

Tip: To restore the DOS prompt to the system default (A>), type the PROMPT command without specifying command-line parameters.

Tip: To enhance your processing capabilities, PROMPT supports the following metacharacters:

Tips and Traps Summary (*continued*)

$b	The | character
$d	The current system date
$e	The ASCII escape character
$g	The greater-than character (>)
$h	The backspace character
$l	The less-than character (<)
$n	The default disk drive
$p	The current disk drive and directory
$q	The equal-sign character (=)
$t	The current system time
$v	The current DOS version number display
$$	The dollar-sign character
$_	The carriage return and linefeed characters

Tip: If you are using DOS subdirectories, set your prompt to [$p]. In so doing, you cause DOS to always display your current directory on the screen display.

Tip: You can use the DOS PROMPT command in conjunction with the ANSI.SYS device driver.

What Is CTTY?

Tip: Only use the DOS CTTY command when you have a terminal (TTY device) connected to your computer's serial port. If you invoke the command

```
A> CTTY AUX
```

without such a device, you will have to reboot your system to continue.

Tips and Traps Summary (*continued*)

Improving Disk-I/O Performance with FASTOPEN

Tip: If you are using a fixed disk under DOS version 3.3, the FAST-OPEN command may improve your system performance by reducing the number of slow disk-I/O operations DOS must perform.

Trap: FASTOPEN supports only fixed disks.

Trap: FASTOPEN installs memory-resident software. As such, you can invoke FASTOPEN only once per computer session. To change FASTOPEN parameters or specify additional disk drives, you must restart DOS.

Trap: Setting the number of FASTOPEN entries too large will slow down system performance.

Using the DOS EXIT Command

Tip: The DOS EXIT command enables you to terminate a secondary command processor.

Using DISKCOMP to Compare Two Floppy Disks

Tip: The DOS DISKCOMP command allows you to compare the contents of two floppy disks.

Trap: DISKCOMP performs a sector-by-sector comparison of two disks. Just because two disks contain the same files does not mean that DISKCOMP will find the disks identical.

Trap: DISKCOMP supports only floppy disks.

Trap: DISKCOMP compares only disks of identical type.

Tips and Traps Summary (*continued*)

Tip: To support your batch processing, DISKCOMP provides the following exit status values:

Exit Status	Meaning
0	Disks compare identically
1	Disks are different
2	User-terminated with CTRL-C
3	Read or write error during comparison
4	Initialization error

Enhancing Character Display with GRAFTABL

Tip: The DOS GRAFTABL command installs memory-resident software that helps the PC display extended characters in medium-resolution graphics mode.

Trap: GRAFTABL loads memory-resident software. Therefore, you can invoke GRAFTABL only once per user session.

Tip: To determine whether GRAFTABL has been loaded, and if so, with which character set, use the /STATUS qualifier to the GRAFTABL command.

File Sharing in a Local-Area Network

Tip: If your computer is not part of a local-area network, do not issue the DOS SHARE command.

Tip: If your computer is part of a local-area network, have the individual who installed the network determine whether the SHARE command is required, and if so, with which settings.

19

International DOS Support

This chapter applies specifically to DOS users abroad who would like to use their country's keyboard template and character set. A few years ago, this chapter would have had a small audience. Today, however, a large number of DOS users are international users. For those of you in the United States, many of you will want to skip ahead to Chapter 20. Instead, it is recommended that you spend a few minutes skimming this chapter. You will be surprised with the capabilities that DOS provides specifically for its international contention.

Using the Correct Character Set

A character set is simply a group of letters and symbols. Throughout this book, you have seen examples of the DOS extended character set and its use in creating files such as

```
A> TYPE FORMAT.NTS

        ┌──────────┐
        │ WARNING! │
        └──────────┘

          FORMAT destroys the contents of your target diskette!

A>
```

Most countries have a unique language; therefore, most countries have a unique character set. DOS version 3.3, as you will see, supports the character sets used by most countries. DOS calls these character sets *codepages.* Unless DOS is told to do otherwise, it loads the character set (codepage) associated with the United States. In this chapter you will learn how to inform your screen display and printer about your desired character set. As you will learn, DOS assigns a unique number to the following codepages.

Codepage	Country
437	United States
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

Tip: *A codepage is simply another name for a character set. By providing codepage support, DOS allows international users to select their desired character set.*

Selecting a Keyboard Template

Just as each country has a unique codepage, most have a specific keyboard layout, or *template.* Depending upon your version of DOS, you will either use the KEYB or KEYBxx command to select your keyboard template.

Tip: *Under DOS version 3.3, the KEYB command allows you to specify one of the following keyboard templates:*

Keyboard Code	Country
BE	Belgium
DK	Denmark
FR	France
GR	Germany
IT	Italy
LA	Latin America
NL	Netherlands
NO	Norway
PO	Portugal
SF	Swiss-French
SG	Swiss-German
SP	Spain
SU	Finland
SV	Sweden
UK	United Kingdom
US	United States

The format of the KEYB command is

```
[drive:][path]KEYB [keyboard_code][,codepage[,template_file]]]
```

where the following is true:

drive: is the disk drive that contains the KEYB.EXE file. If you do not specify a disk drive letter, DOS will use the current default.

path is the subdirectory that contains the KEYB.EXE file. If you do not specify a subdirectory, DOS uses the current default directory.

keyboard_code is the 2-letter identifier for the desired keyboard, shown previously.

codepage specifies the codepage that defines the character set.

template_file specifies the location of the KEYBOARD.SYS file.

Tip: *The KEYBOARD.SYS file contains the keyboard templates for the international keyboards.*

Assuming, for example, that the KEYBOARD.SYS file resides in the current directory

```
A> DIR  KEYBOARD.SYS

Volume in drive C is DOSDISK
Directory of  C:\

KEYBOARD        SYS        19735   7-24-87  12:00a
```

most users will invoke the KEYB command, as shown here:

```
A> KEYB  GR
```

In this case, the German keyboard template was selected. If KEYBOARD.SYS resides in a DOS subdirectory, specify the subdirectory name in the KEYB command, as shown here:

```
C> KEYB  GR,,\DOS\KEYBOARD.SYS
```

Note that the command does not specify the optional codepage. Still, you must use the two commas together as shown to inform KEYB.

Tip: *DOS versions prior to DOS 3.3 provide several different keyboard templates in individual files, as shown here:*

File	Country
KEYBFR.COM	France
KEYBGR.COM	Germany
KEYBIT.COM	Italy
KEYBSP.COM	Spain
KEYBUK.COM	United Kingdom

Using these files, to select the German keyboard template, you would use the command

```
A> KEYBGR
```

Regardless of which method you use to install the keyboard template, DOS allows you to switch quickly between the foreign template and the U.S. keyboard.

Tip: *The CTRL-ALT-F2 key combination selects the foreign keyboard template. Likewise, the CTRL-ALT-F1 key combination selects the U.S. keyboard.*

By using CTRL-ALT-F2 and CTRL-ALT-F1, you can toggle back and forth between the U.S. and foreign keyboards as your needs require.

Tip: *When using foreign keyboard templates, some keys may appear to have no effect. These keys are called dead keys.*

To generate the symbol that corresponds to a *dead key,* you must hold down the CTRL and ALT keys while pressing the dead key.

What Is a Symbol Set?

In addition to having a unique keyboard template and character set, most countries use unique symbols for currency, as well as different date and time formats. The U.S. format for the date, for example, is mm-dd-yy. However, some countries use dd-mm-yy.

To allow use of the symbols and formats specific to another country, DOS provides the COUNTRY= entry in CONFIG.SYS.

Tip: *The COUNTRY= entry in CONFIG.SYS allows you to select the specific date, time, and currency formats for your country at system startup.*

Prior to DOS version 3.3, the format of the COUNTRY= entry was

```
COUNTRY=country_code
```

For DOS version 3.3, however, the format becomes

```
COUNTRY=country_code [,codepage [,country_file]]
```

where the following is true:

country_code is the 3-digit value that corresponds to the following countries:

Country Code	Country
001	United States
002	French-Canadian
003	Latin America
031	Netherlands
032	Belgium
033	France
034	Spain

Country Code	Country
039	Italy
041	Switzerland
044	United Kingdom
045	Denmark
046	Sweden
047	Norway
049	Germany
061	English
351	Portugal
358	Finland
785	Arabic countries
972	Israel

codepage specifies the desired codepage.

country_file specifies the location of the COUNTRY.SYS file.

Assuming, for example, that the COUNTRY.SYS file resides in the DOS subdirectory, your COUNTRY= entry might be

```
COUNTRY=049,,\DOS\COUNTRY.SYS
```

In this case, the entry selects the German symbol set.

What You Need to Get Started

If you plan to use only one character set and one keyboard template, regardless of which country you desire, to get started with the foreign codepage and keyboard template is easy. First, place the correct COUNTRY= entry in CONFIG.SYS. This example will install support for Germany:

```
COUNTRY=049,,\DOS\COUNTRY.SYS
```

Second, place the desired KEYB command in AUTOEXEC.BAT:

```
KEYB GR,,\DOS\KEYBOARD.SYS
```

Once you do so, you need to restart DOS in order for the changes to take effect.

Multiple-Language Users

Some users have to work with DOS under varying keyboard templates and different character sets. Those users must direct their systems to perform codepage switching. As you will see, changing from one codepage to another is quite straightforward as long as you perform a few simple steps.

Before you can switch from one code to another with your screen display, you must install the DISPLAY.SYS device driver in CONFIG.SYS.

Tip: *The DISPLAY.SYS device driver provides codepage switching for EGA, LCD, and VGA monitors.*

The format of the DISPLAY.SYS entry is

```
DEVICE=DISPLAY.SYS CON:=[(display_type[,hardware_codepage]
[,(additional_codpages, subfonts)])]
```

where the following is true:

display_type specifies your screen display type. Valid entries include MONO, CGA, EGA, and LCD.

hardware_codepage specifies the codepages supported by your screen display. Valid entries include 437, 850, 860, 863, and 865.

additional_codepages specifies the number of pages that DOS can prepare for use. MONO and CGA systems do not support codepage switching and must be set to zero. EGA can be set to 2, while LCD can be set to 1.

subfonts specifies the number of subfonts supported by the display.

To install DISPLAY.SYS for an EGA monitor, for example, your CONFIG.SYS entry might be

```
DEVICE=DISPLAY.SYS CON:=(EGA,437,(2,1))
```

Likewise, to provide codepage switching support for your printer, you must install the PRINTER.SYS device driver in CONFIG.SYS.

Tip: *The PRINTER.SYS device driver provides codepage switching support for the IBM ProPrinter and QuietWriter printers.*

The format of the PRINTER.SYS entry is

```
DEVICE=PRINTER.SYS LPTn:=[(printer_type[,(hardware_codepage)
        additional_codepages])]]
```

where the following is true:

LPTn specifies the desired printer port (LPT1, LPT2, or LPT3).

printer_type specifies the printer type: 4201 for the IBM ProPrinter, and 5202 for the QuietWriter.

hardware_codepage specifies the codepage supported by your printer. Valid values include 437, 850, 860, 863, and 865.

additional_codepages specifies the number of codepages that DOS can prepare for your printer. This value is specific to this printer.

To install the PRINTER.SYS device driver for a QuietWriter printer, use the entry

```
DEVICE=PRINTER.SYS LPT1:=(4201,863,1)
```

Remember that once you install either of these device drivers, you must reboot DOS for the changes to take effect.

Next, you must invoke the DOS NLSFUNC command.

Tip: *NLSFUNC is the DOS national language support function. You must invoke NLSFUNC before DOS will perform codepage switching.*

The format of the NLSFUNC command is

```
NLSFUNC [country_file]
```

where *country_file* specifies the location of the COUNTRY.SYS file.

Once you invoke NLSFUNC, your system is ready to perform codepage switching. To switch between two codepages, use the CHCP command.

Tip: *The DOS CHCP command allows you to select a new codepage for your screen display and printer.*

The format of the CHCP command is

```
CHCP [codepage]
```

where *codepage* specifies the desired codepage:

Codepage	Country
437	United States
850	Multilingual
860	Portuguese

Codepage	Country
863	French-Canadian
865	Nordic

Tip: *If you invoke CHCP without specifying a codepage, CHCP will display the current codepage.*

At the DOS prompt, enter

```
A> CHCP
```

DOS displays the current codepage, as shown here:

```
A> CHCP

Active code page:  437
```

In this case, CHCP displays the fact that the system is using the default United States codepage (437).

If you attempt to select a different codepage,

```
A> CHCP  865
```

the command will fail and DOS will display

```
A> CHCP  865

Code page  865 not prepared for system

A>
```

Trap: *Before CHCP can select a codepage, you must prepare the codepage for use by the system with the DOS MODE command.*

Earlier in this book you found that MODE allows you to set several device-specific characteristics. As you will see here, MODE is a key component in preparing and selecting codepages.

Using Mode For Codepage Support

Tip: *DOS provides several codepage information (CPI) files on your DOS disks.*

```
EGA     CPI   49065   7-24-87  12:00a
LCD     CPI   10752   7-24-87  12:00a
4201    CPI   17089   7-24-87  12:00a
5202    CPI   17459   7-24-87  12:00a
```

You will need these files as you prepare your system for codepage switching.

To prepare your screen display for codepage switching, use the following MODE command format:

```
MODE CON CODEPAGE PREPARE=(codepage,codepage) display_type.cpi
```

For example, to prepare codepages for an EGA monitor, your entry might be

```
C> MODE CON CODEPAGE PREPARE=(860,865) EGA.CPI
```

Once you prepare a codepage for your screen display, you can select the codepage using MODE as will be discussed in a moment. In a similar manner, to prepare a codepage for your printer, use the MODE format

```
MODE LPTn CODEPAGE PREPARE=(codepage) printer.cpi
```

where *LPTn* specifies the printer port, and *printer.cpi* is either 4201.CPI or 5202.CPI.

```
C> MODE LPT1 CODEPAGE PREPARE=(863) 5202.CPI
```

Tip: *CHCP selects a codepage for your printer and screen in one command. If you want to change the codepage for your printer only or for your screen display only, use the MODE command to select the codepage for the device.*

In addition to letting you prepare codepages for a device, MODE allows you to select a codepage for a device. The format of the MODE command in this case is

```
MODE device CODEPAGE SELECT=codepage
```

For example, this command selects the codepage 865 for your screen display, leaving the printer codepage unchanged:

```
C> MODE CON CODEPAGE SELECT=865
```

Trap: *If the device for which you have prepared a codepage gets turned off, it will forget the codepage it was using and revert to the default codepage.*

If, for example, your printer is turned off after you have prepared a codepage for it, it will resume printing with the default codepage. To remind the printer that you are using a different codepage, issue the REFRESH option of the MODE command:

```
C> MODE LPT1 CODEPAGE REFRESH
```

DOS will restore the current codepage to the printer.

Tip: *To display the current codepages that are prepared for a specific device, use the MODE /STATUS qualifier.*

If you are not sure which codepages have been prepared for a device, use the /STATUS qualifier, as shown here:

```
Active codepage for device CON is 437
hardware codepages:
harCodepage 437
prepared codepages:
haCodepage 860
haCodepage 865
MODE Status Codepage function completed
```

Tip: *To simplify your codepage processing, MODE supports the following abbreviations:*

CODEPAGE	CP
PREPARE	PREP
SELECT	SEL
REFRESH	REF
/STATUS	/STA

Using these abbreviations, the following commands are functionally equivalent:

```
A> MODE CON CODEPAGE PREPARE=(860) EGA.CPI
```

and

```
A> MODE CON CP PREP=(860) EGA.CPI
```

```
A> MODE CON CODEPAGE SELECT=860
```

and

```
A> MODE CON CP SEL=860
```

Keep in mind that the only time you need to issue these commands is when you want to switch from one codepage to another. Most international users will simply want to select one codepage and keyboard template.

Using the DOS SELECT Command

To support floppy-disk-based international systems, the DOS SELECT command helps you create a bootable floppy disk containing the correct COUNTRY= entry in CONFIG.SYS as well as the correct KEYB command in AUTOEXEC.BAT. By this stage in the book, you should be fully capable of editing the CONFIG.SYS and AUTOEXEC.BAT files to insert your own country-specific entries:

```
A> COPY CON CONFIG.SYS
FILES=20
BUFFERS=25
COUNTRY=049
^Z
            1 File(s) copied
A> COPY CON AUTOEXEC.BAT
KEYB GR
^Z
            1 File(s) copied
```

The DOS SELECT command essentially provides this function for you. For more information on SELECT, refer to the command reference section of this book.

Tips and Traps Summary

Using the Correct Character Set

Tip: A codepage is simply another name for a character set. By providing codepage support, DOS allows international users to select their desired character set.

Selecting a Keyboard Template

Tip: Under DOS version 3.3, the KEYB command allows you to specify one of the following keyboard templates:

Keyboard Code	Country
BE	Belgium
DK	Denmark
FR	France
GR	Germany
IT	Italy
LA	Latin America
NL	Netherlands
NO	Norway
PO	Portugal
SF	Swiss-French
SG	Swiss-German
SP	Spain
SU	Finland
SV	Sweden
UK	United Kingdom
US	United States

Tips and Traps Summary (*continued*)

Tip: The KEYBOARD.SYS file contains the keyboard templates for the international keyboards.

Tip: DOS versions prior to DOS 3.3 provide several different keyboard templates in individual files, as shown here:

File	Country
KEYBFR.COM	France
KEYBGR.COM	Germany
KEYBIT.COM	Italy
KEYBSP.COM	Spain
KEYBUK.COM	United Kingdom

Tip: The CTRL-ALT-F2 key combination selects the foreign keyboard template. Likewise, the CTRL-ALT-F1 key combination selects the U.S. keyboard.

Tip: When using foreign keyboard templates, some keys may appear to have no effect. These keys are called dead keys.

What Is a Symbol Set?

Tip: The COUNTRY= entry in CONFIG.SYS allows you to select the specific date, time, and currency formats for your country at system startup.

Multiple-Language Users

Tip: The DISPLAY.SYS device driver provides codepage switching for EGA, LCD, and VGA monitors.

Tip: The PRINTER.SYS device driver provides codepage switching support for the IBM ProPrinter and QuietWriter printers.

Tips and Traps Summary (*continued*)

Tip: NLSFUNC is the DOS national language support function. You must invoke NLSFUNC before DOS will perform codepage switching.

Tip: The DOS CHCP command allows you to select a new codepage for your screen display and printer.

Tip: If you invoke CHCP without specifying a codepage, CHCP will display the current codepage.

Trap: Before CHCP can select a codepage, you must prepare the codepage for use by the system with the DOS MODE command.

Using Mode for Codepage Support

Tip: DOS provides several codepage information (CPI) files on your DOS disks.

Tip: CHCP selects a codepage for your printer and screen in one command. If you want to change the codepage for your printer only or for your screen display only, use the MODE command to select the codepage for the device.

Trap: If the device for which you have prepared a codepage gets turned off, it will forget the codepage it was using and revert to the default codepage.

Tip: To display the current codepages that are prepared for a specific device, use the MODE /STATUS qualifier.

Tip: To simplify your codepage processing, MODE supports the following abbreviations:

CODEPAGE	CP
PREPARE	PREP
SELECT	SEL
REFRESH	REF
/STATUS	/STA

20

Using BACKUP and RESTORE

Trap: *If you do not back up your disks on a regular basis, you are essentially playing Russian roulette with your files. Eventually, you will lose information or programs.*

Whether your system is floppy-disk-based or uses a large fixed disk, you must take precautions with your files. Earlier in this book you learned about write protecting your disks, never using your original disks, and several rules for handling your disks. In Chapter 16 you learned that the DOS ATTRIB command can protect your files from errant DOS commands such as

```
A> DEL *.*
```

All of these precautions will help you prevent loss of data. However, without backup disks, you will have no protection from a lost disk, a damaged disk, theft, or other catastrophic events. Unfortunately, lost or damaged disks are much too common.

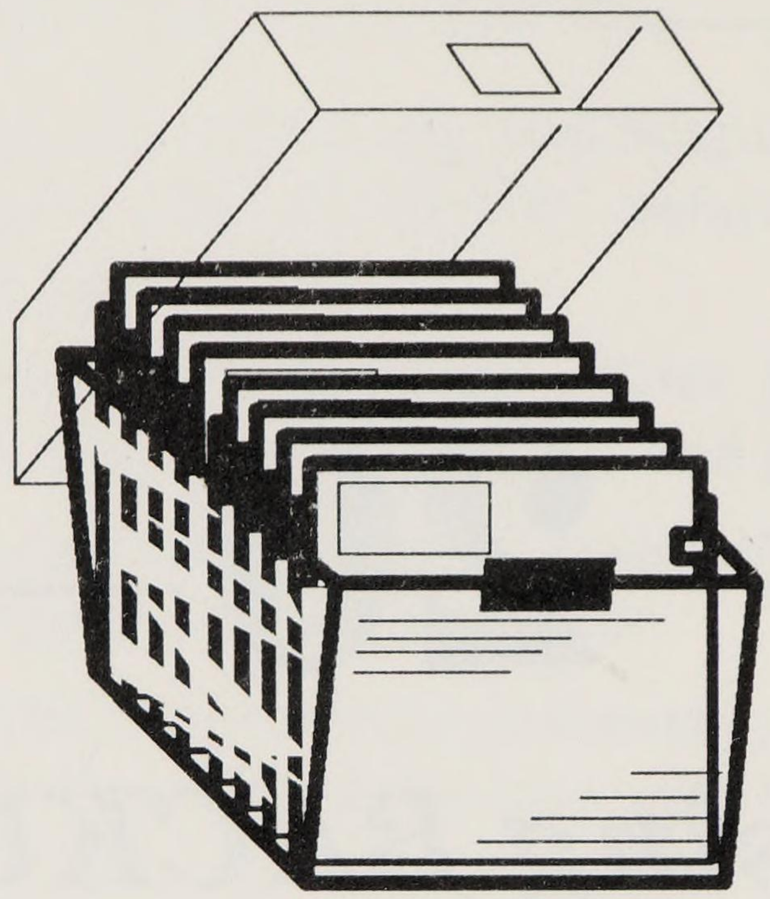


Figure 20-1. A disk storage container

This chapter discusses how to duplicate your disks and files to prevent loss of programs or information. Specifically, this chapter will examine the BACKUP and RECOVER commands, which should become part of your daily procedures.

Floppy-Disk-Based Systems

Even if your system does not have a fixed disk, backing up your files is still an important process that you need to perform on a daily basis. Earlier in this book you learned how to use the DOS DISKCOPY command to make duplicate copies of your original disks. You should store all of your original disks safely in a disk storage container (see Figure 20-1) and place the container in a room other than your computer room. In so doing, you prevent such events as spills or theft from destroying your working disks and originals at the same time.

Keep track of the files and disks that you modify throughout the day. Spend the last few minutes of each day copying those files to a backup disk that you can place in a safe location. At the end of the week, consolidate your files into one of the disks that you can store for a longer period of time.

When to Use BACKUP

The DOS BACKUP command exists soley to assist you in backing up your files. Although BACKUP supports floppy-disk-based systems, it is most effective when used to back up a fixed disk to one or more floppy disks. Although fixed disks give you tremendous storage capabilities, they also give you a single point of failure. If your fixed disk becomes damaged, or if you issue an errant command such as

```
C> FORMAT C:
```

all of the files on your disk can be lost. Therefore, the need to back up a fixed disk should be apparent.

Tip: *BACKUP allows you to back up specific files, the entire disk, or files meeting specific conditions.*

As you will see later in this chapter, BACKUP allows you to easily back up one or more files, or even an entire disk. In addition to these common operations, BACKUP allows you to back up files based on the following characteristics.

- Creation date: BACKUP will back up only the files created or modified on or since the date specified.
- Creation time: BACKUP will back up only the files created or modified on or after the time specified.
- Archive attribute set: BACKUP will back up only those files with their archive attribute set. In doing so, BACKUP is backing up only those files created or modified since the last BACKUP command.

As discussed, when BACKUP backs up the files on your fixed disk to floppies, it will create one or more floppy disks (see Figure 20- 2).

Trap: *BACKUP places files on the floppy disk in a unique format that only the DOS BACKUP and RESTORE commands understand.*

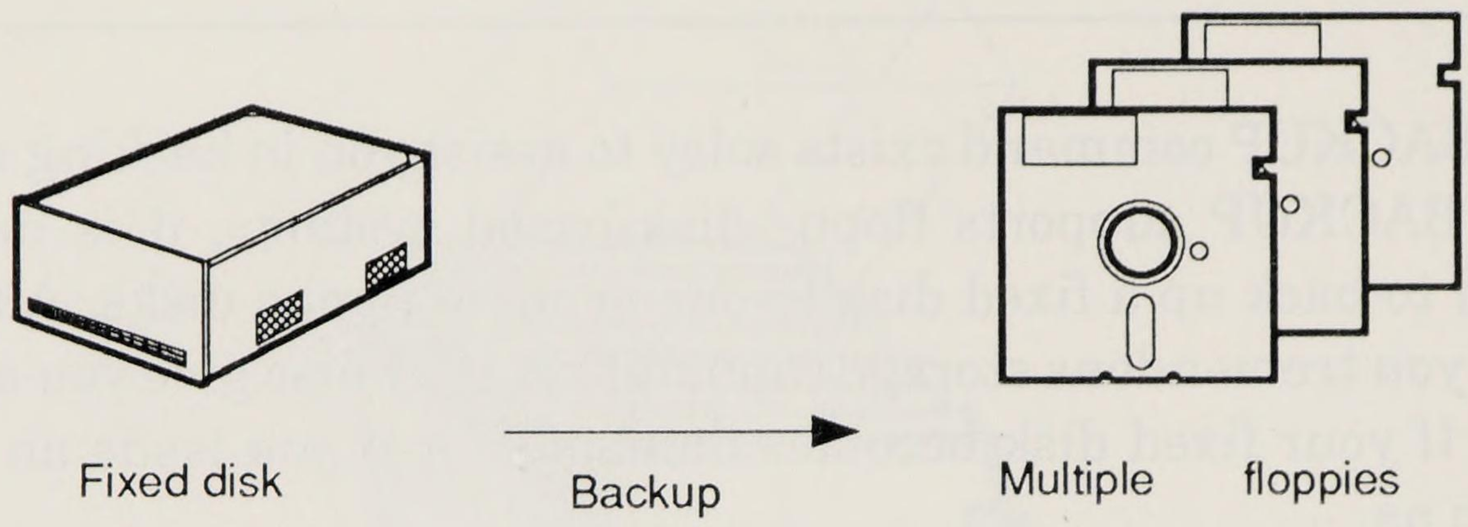


Figure 20-2. BACKUP backs up files from your fixed disk to floppy disks

If you ever need to obtain the backup copy of a file from your backup disks, the only way to copy the file back to your fixed disk is with the DOS RESTORE command. If you simply copy the file from the backup disk, the file will be invalid because BACKUP places information in front of the file that the RESTORE command removes when it places the file back onto your fixed disk. Just as the DOS BACKUP command copies files from your fixed disk to floppies, the DOS RESTORE command places selected files back onto your fixed disk as shown in Figure 20-3.

Getting Started with BACKUP

With your Supplemental Programs Disk in drive A, issue these commands:

```
A> DIR  BACKUP.COM

Volume in drive A is MS330PP02
Directory of  A:\

BACKUP        COM        29976   7-24-87  12:00a
            1 File(s)     2048 bytes free

A> DIR  RESTORE.COM
```

```
Volume in drive A is MS330PP02
Directory of  A:\

RESTORE      COM        35650   7-24-87  12:00a
          1 File(s)     2048 bytes free
```

As you can see, BACKUP and RESTORE are external DOS commands. The next few sections discuss the use of BACKUP and RESTORE in conjunction with a fixed disk. If your system is floppy-disk- based, you might want to skim this section for your own information.

The format of the DOS BACKUP command is

```
[drive:][path]BACKUP [source] [target_drive] [/A] [/F]
[/D:mm-dd-yy] [/L:logfile] [/M] [/S] [/T:hh:mm:ss]
```

where the following is true:

drive specifies the location of the BACKUP.COM file. If you do not specify a disk drive, DOS uses the current default.

path specifies the location of the DOS subdirectory that contains the BACKUP.COM file. If you do not specify a disk drive, DOS uses the current default.

source specifies the file(s) to be backed up. BACKUP fully supports the DOS wildcard characters. You must specify a source disk drive.

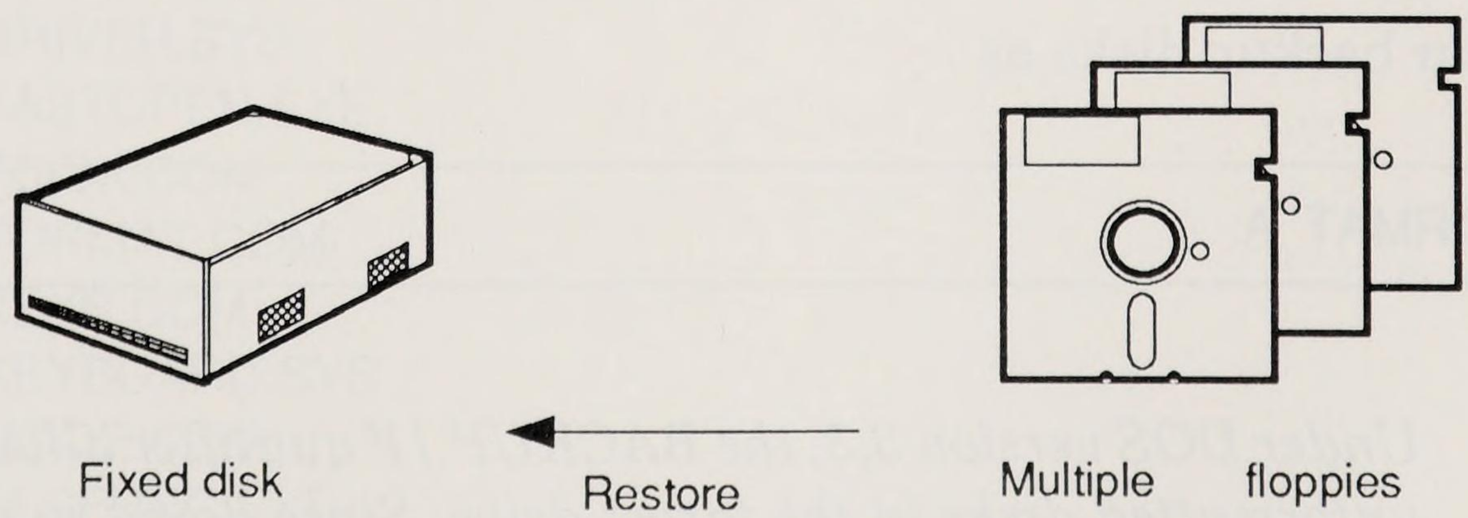


Figure 20-3. RESTORE places files onto your fixed disk from floppy disks

target_drive specifies the disk drive that the files are backed up to.

/A directs BACKUP to add the files to the backup files that already exist on the target drive.

/F is a DOS version 3.3 qualifier that directs BACKUP to format the target disk before performing the backup operation.

/D:mm-dd-yy directs BACKUP to back up only those files created or modified on or after the specified date.

/L:logfile is a DOS version 3.3 qualifier that directs DOS to create a *log file* that specifies which files exist on each backup disk.

/M directs BACKUP to back up only those files created or modified since the last BACKUP command.

/S directs BACKUP to back up also those files that reside in subdirectories below the source.

/T:hh:mm:ss directs BACKUP to back up only those files that have been created or modified since the last BACKUP command.

To use the examples throughout this chapter, have two formatted disks available.

Trap: *Do not use the DOS FORMAT /S command to format your backup disks. In so doing, the system files that DOS places on the disk would consume considerable space that DOS could otherwise utilize.*

Format your backup disks as

```
C> FORMAT A:
```

Tip: *Under DOS version 3.3, the BACKUP /F qualifier allows you to use unformatted disks in the target drive. Since doing so increases the amount of time BACKUP requires, always try instead to have sufficient formatted disks available.*

If BACKUP takes too long, you might use that as an excuse not to perform system backups. To reduce the amount of time that BACKUP requires, use formatted disks.

With one of your formatted disks in drive A, issue the command

```
C> BACKUP  C:\DOS\*.*  A:
```

This command assumes that your DOS commands reside in a subdirectory called DOS on your fixed disk. If this is the case, BACKUP will display the message

```
Insert backup diskette 01 in drive A:

Warning! Files in the target drive
A:\ root directory will be erased
Strike any key when ready
```

BACKUP is prompting you to insert a backup disk in drive A, warning you that if files exist in the target directory, BACKUP will overwrite them. If you have not already done so, place a formatted disk in drive A and press ENTER. BACKUP will begin backing up your files, displaying

```
*** Backing up files to drive A: ***
Diskette Number: 01

\DOS\COMMAND.COM
\DOS\ANSI.SYS
\DOS\COUNTRY.SYS
\DOS\DISPLAY.SYS
\DOS\DRIVER.SYS
\DOS\FASTOPEN.EXE
\DOS\FDISK.COM
\DOS\FORMAT.COM
\DOS\KEYB.COM
\DOS\KEYBOARD.SYS
\DOS\MODE.COM
\DOS\NLSFUNC.EXE
\DOS\PRINTER.SYS
\DOS\REPLACE.EXE
\DOS\SELECT.COM
```

```
\DOS\SYS.COM
\DOS\VDISK.SYS
\DOS\XCOPY.EXE
\DOS\EGA.CPI
\DOS\LCD.CPI
\DOS\4201.CPI
\DOS\5202.CPI
\DOS\APPEND.EXE
\DOS\ASSIGN.COM
\DOS\ATTRIB.EXE
\DOS\BACKUP.COM
\DOS\BASIC.COM
\DOS\BASICA.COM
```

If you are using a 1.2MB disk, the BACKUP command will complete. However, if you are using a 360K disk instead, the disk will fill and BACKUP will prompt you to insert a second disk, as shown here:

```
Insert backup diskette 02 in drive A:

Warning! Files in the target drive
A:\ root directory will be erased
Strike any key when ready
```

Tip: *If BACKUP requires more than one disk to perform a backup operation, it will prompt you to insert new disks as required.*

When the BACKUP command completes, issue the command

```
C> BACKUP  C:\DOS\*.*  A:  /M
```

The /M qualifier directs BACKUP to back up only those files created or modified since the last BACKUP command. As before, BACKUP will prompt

```
Insert backup diskette 01 in drive A:

Warning! Files in the target drive
A:\ root directory will be erased
Strike any key when ready
```

Simply insert one of your disks from earlier. Don't worry if you just used the disk for the previous BACKUP command—you will overwrite the disk's contents several times in this chapter. When you press ENTER, BACKUP will display

```
Warning! No files were found to back up
```

Since you have not created or modified any files in the DOS subdirectory since the last BACKUP command, BACKUP found no files to back up. Later in this chapter you will learn how to use BACKUP to back up only the files that you have created that day. At that time, you will again use the /M qualifier in the BACKUP command line.

Next, issue the command

```
C> BACKUP  C:\DOS\*.COM  A:
```

In this case, the processing becomes

```
*** Backing up files to drive A: ***
Diskette Number: 01

\DOS\COMMAND.COM
\DOS\FDISK.COM
\DOS\FORMAT.COM
\DOS\KEYB.COM
\DOS\MODE.COM
\DOS\SELECT.COM
\DOS\SYS.COM
\DOS\ASSIGN.COM
\DOS\BACKUP.COM
\DOS\BASIC.COM
\DOS\BASICA.COM
\DOS\CHKDSK.COM
\DOS\COMP.COM
\DOS\DEBUG.COM
\DOS\DISKCOMP.COM
\DOS\DISKCOPY.COM
\DOS\EDLIN.COM
\DOS\GRAFTABL.COM
\DOS\GRAPHICS.COM
```

```
\DOS\LABEL.COM
\DOS\MORE.COM
\DOS\PRINT.COM
\DOS\RECOVER.COM
\DOS\RESTORE.COM
\DOS\TREE.COM
```

When that command completes, issue the command

```
C> BACKUP C:\DOS\*.SYS A: /A
```

The /A qualifier directs BACKUP to add the files it is backing up to those files already residing on the backup disk. Therefore, BACKUP prompts you to

```
Insert last backup diskette in drive A:
Strike any key when ready
```

By using the /A qualifier, you can continue to add backup files to the disk until the disk becomes full. When the disk fills, DOS will prompt you to enter a second disk, as you have seen before:

```
Insert backup diskette 02 in drive A:

Warning! Files in the target drive
A:\ root directory will be erased
Strike any key when ready
```

So that you will better understand the DOS RESTORE command, create the TEMP subdirectory and the ONE.DAT, TWO.DAT, and THREE.DAT files, as shown here:

```
C> MKDIR \TEMP
C> COPY CON \TEMP\ONE.DAT
1
^Z
        1 File(s) copied

C> COPY CON \TEMP\TWO.DAT
2
^Z
        1 File(s) copied
```

```
C> COPY  CON  \TEMP\THREE.DAT
3
^Z
            1 File(s) copied

C>
```

Next, issue the command

```
C> BACKUP  C:\TEMP\*.*   A:
```

As before, BACKUP will display

```
Insert backup diskette 01 in drive A:

Warning! Files in the target drive
A:\ root directory will be erased
Strike any key when ready
```

When the command completes, delete the files that TEMP contains and remove the directory:

```
C> DEL  \TEMP\*.*
Are you sure (Y/N)? Y

C> RMDIR  \TEMP
```

To get the files back, you must use the DOS RESTORE command.

Using the RESTORE Command

As discussed, BACKUP and RESTORE work in opposite manners. The format of the DOS RESTORE command is

```
[drive:][path]RESTORE source_drive: [files] [/A:mm-dd-yy]
[/Bmm-dd-yy][E:hh:mm:ss] [/L:hh:mm:ss] [/M]
[/N] [/P] [/S]
```

where the following is true:

drive specifies the disk drive containing the RESTORE.COM file. If you do not specify a disk drive, DOS uses the current default.

path specifies the DOS subdirectory that contains the RESTORE.COM file. If you do not specify a disk drive, DOS uses the current directory.

source_drive specifies the disk drive letter of the disk containing the backup disks.

files specifies the files to restore. RESTORE fully supports the DOS wildcard characters.

/*A:mm-dd-yy* directs RESTORE to restore only those files created or modified on or after the specified date.

/*B:mm-dd-yy* directs RESTORE to restore only those files created or modified on or before the specified date.

/*E:hh:mm:ss* directs RESTORE to restore only those files created or modified on or before the specified time.

/*L:hh:mm:ss* directs RESTORE to restore only those files created or modified on or after the specified time.

/*M* directs RESTORE to restore only those files modified since the last BACKUP operation.

/*N* directs RESTORE to restore only those files that no longer reside on the target disk.

/*P* directs RESTORE to prompt

```
Warning! File FILENAME.EXT
is a read-only file
Replace the file (Y/N)?
```

with each file that has been marked read-only, or

```
Warning! File FILENAME.EXT
was changed after it was backed up
Replace the file (Y/N)?
```

with each file that has changed since the BACKUP command.

/*S* directs RESTORE to restore files that are contained in subdirectories below the source.

To restore the ONE.DAT, TWO.DAT, and THREE.DAT files that you just deleted, use the command

```
C> RESTORE A: C:\TEMP\*.*
```

In this case, RESTORE will prompt

```
Insert backup diskette 01 in drive A:
Strike any key when ready
```

Place the disk containing backup files in drive A and press ENTER. RESTORE will continue displaying

```
Insert backup diskette 01 in drive A:
Strike any key when ready

*** Files were backed up 07-10-1988 ***

*** Restoring files from drive A: ***
Diskette: 01
\TEMP\ONE.DAT
\TEMP\TWO.DAT
\TEMP\THREE.DAT
```

A directory listing of the TEMP subdirectory shows that the files have been restored as desired. Later in this chapter you will learn how to restore all of the files on a disk.

Labeling Your Disks

A key component to backup success is correctly labeling your disks. Most users will want to use a format similar to that shown in Figure 20-4. Remember, never write on your disk label once you have placed the label on your disk.

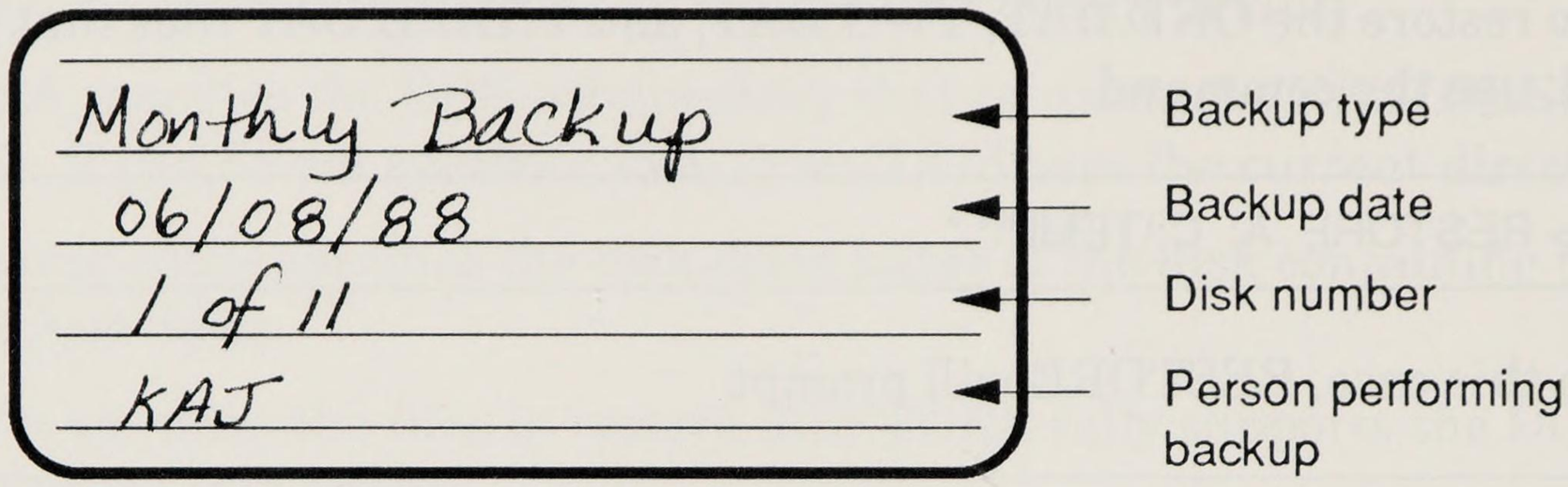


Figure 20-4. Label for disk as BACKUP completes

Developing a Backup Policy

To simplify your file backup operations, you should develop a set of backup procedures that you will perform on a daily basis. Before doing so, however, you must understand the differences between monthly disk backups and daily incremental backups.

The first day of each month, you should back up *every* file on your fixed disk. To do so, you must determine the number of floppy disks that the backup operation will require.

Tip: *Use the CHKDSK command to determine the amount of space that your disk is using.*

To determine the amount of disk space in use, invoke CHKDSK.

```
C> CHKDSK

Volume DOSDISK    created Jun 17, 1988 3:49a

33439744 bytes total disk space
53248 bytes in 3 hidden files
202752 bytes in 87 directories
23197696 bytes in 2460 user files
14336 bytes in bad sectors
9971712 bytes available on disk

655360 bytes total memory
587488 bytes free
```

Next, use the equation

bytes in use = total disk space / bytes available

In this case, the equation becomes

bytes in use = 33,439,744 - 9,971,712
= 23,468,032

Using Table 20-1, determine the number of floppy disks that your backup operation will require, using the equation

number of floppies = bytes in use / storage capacity

In this case, the equation becomes

number of floppies = 23,468,032 / 1,228,800
= 19.09
= 20 1.2MB floppies

Once you know (and have available) the number of formatted disks, issue the command

```
C> BACKUP  C:\*.*  A:  /S
```

Table 20-1. Number of Bytes Available on Different Types of

Disk Type	Storage Capacity
Single-sided Double-density	184,320 bytes
Double-sided Double-density	368,640 bytes
Double-sided Double-density 3.5	737,280 bytes
Quad-density	1,228,800 bytes
High-capacity 3.5	1,474,560 bytes

This command will back up every file on your fixed disk. Because it may take a considerable amount of time, you have to perform this command only once a month.

Each day, for the remainder of the month, you will back up only the files that you have created that day, by using the command

```
C> BACKUP C:\*.* A: /S /A /M
```

Your daily backup operations should require only a few minutes of your time each day.

In detail, here is how your monthly backup procedures work: Purchase enough disks to perform *two* complete disk backups; next, obtain a disk media storage box, shown earlier, to specifically store your backup disks. On the first day of the month, enter the command

```
C> BACKUP C:\*.* A: /S
```

to back up your entire fixed disk. Label each of your monthly backup disks as shown in Figure 20-5. Place your monthly disks in a safe location—you won't need them for another month. Each day for the rest of the month, using a different set of disks, issue the command

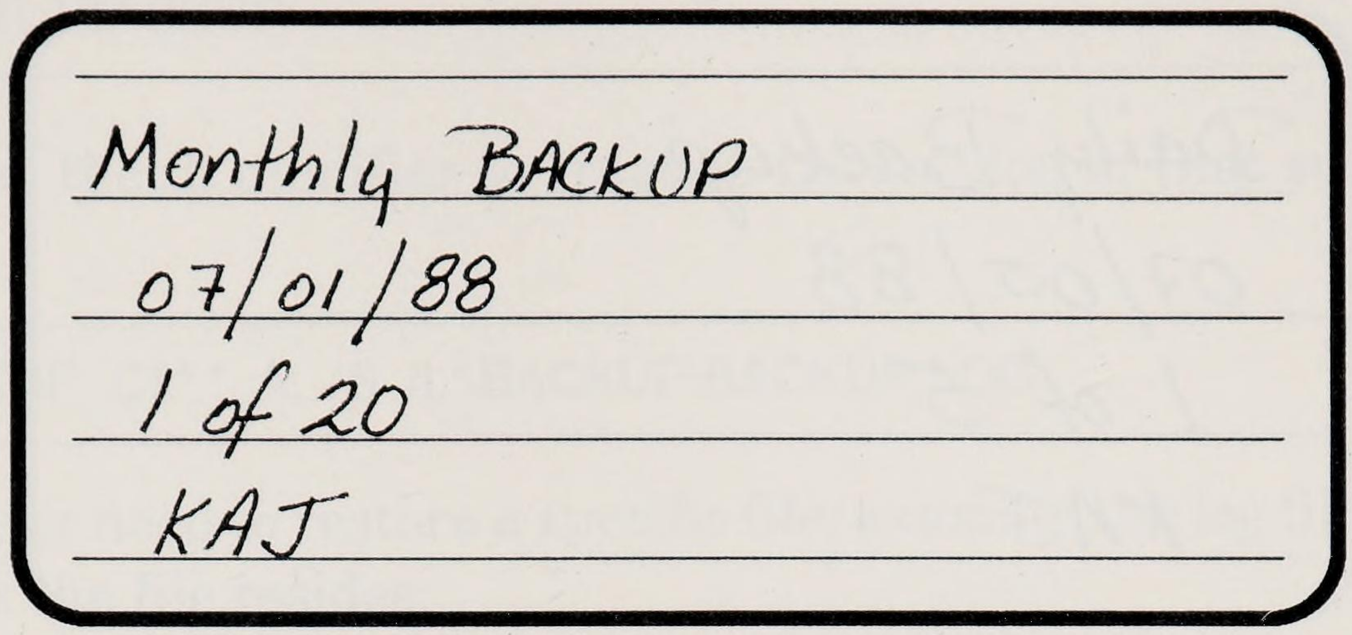


Figure 20-5. Disk label for monthly backup

```
C> BACKUP C:\*.* A: /A /S /M
```

which backs up only the files that you have created since the previous BACKUP command (which hopefully occurred the day before). Since you are using the /A qualifier, BACKUP will add files to your backup disk until it fills. Therefore, you will likely be able to use the same disk for several days. When BACKUP prompts

```
Insert backup diskette 02 in drive A:

Warning! Files in the target drive
A:\ root directory will be erased
Strike any key when ready
```

insert a new disk into your disk drive and continue. In this case, label your disks as shown in Figure 20-6. When the first day of the next month occurs, again issue the command

```
C> BACKUP C:\*.* A: /S
```

If this command completes successfully, you can recycle the previous month's backup disks, using them for this month's backup files.

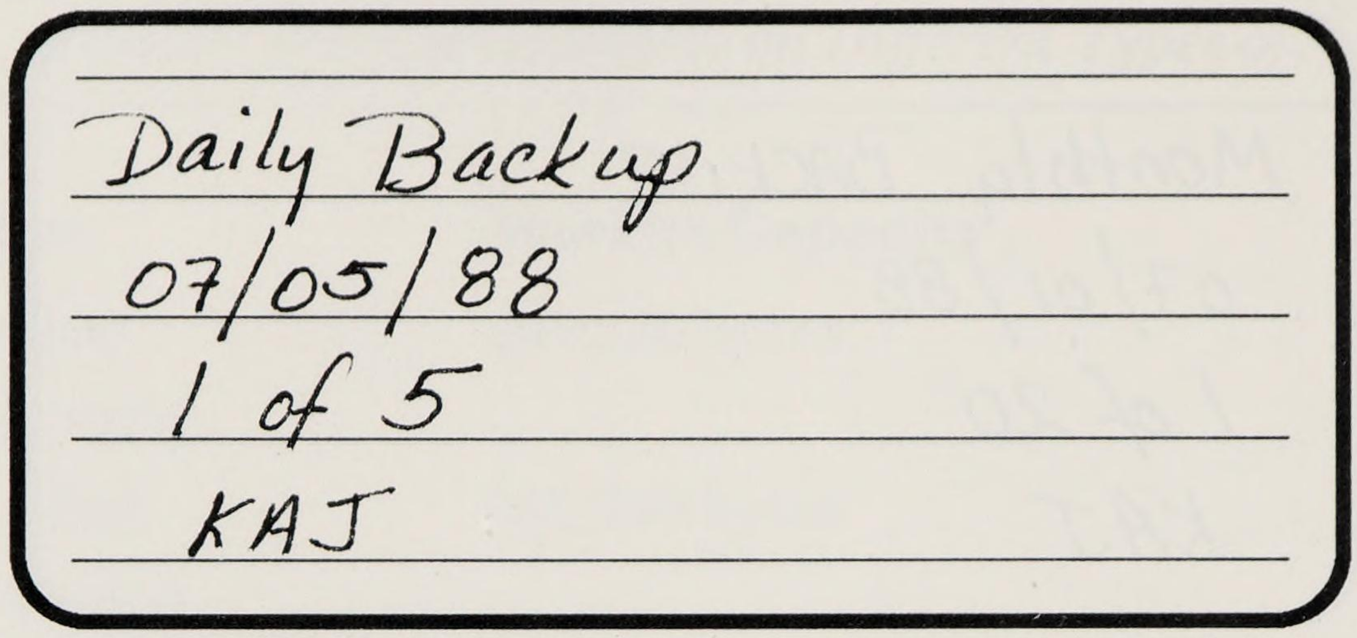


Figure 20-6. Disk label for daily backup

What Is a Log File?

Once you back up files from a fixed disk to floppies, the files are available on disk should you ever need to restore them. The difficulty, however, if you need to do so, is in locating the disk containing the files that you need to restore among your collection of backup disks. If you are using DOS version 3.3, the /L:logfile qualifier directs BACKUP to create a file containing a list of the files it backs up along with the disks on which they reside.

Tip: *Under DOS version 3.3, the /L qualifier directs BACKUP to create a log file, which contains a list of the files that DOS has placed on each disk.*

To use the /L qualifier, specify the name of the *log file* you want DOS to create, as shown here:

```
C> BACKUP C:\*.* A: /S /L:C:\BACKUP.LOG
```

If you are using DOS version 3.3, create a subdirectory on your fixed disk called BACKUP.

```
C> MKDIR \BACKUP
```

Next, use the /L qualifier to create a backup log in that subdirectory, as shown here:

```
C> BACKUP C:\*.* A: /S /L:\BACKUP\BACKUP.LOG
```

If you later need to restore a specific file, examine the log file to determine on which disk the file resides.

```
C> PRINT \BACKUP\BACKUP.LOG
```

In this case, the BACKUP.LOG file might contain

```
7-10-1988 8:59:37
001 \IBMBIO.COM
001 \IBMDOS.COM
001 \COMMAND.COM
001 \AUTOEXEC.BAT
001 \CONFIG.SYS
001 \DOS\COMMAND.COM
001 \DOS\ANSI.SYS
001 \DOS\COUNTRY.SYS
001 \DOS\DISPLAY.SYS
001 \DOS\DRIVER.SYS
001 \DOS\FASTOPEN.EXE
001 \DOS\FDISK.COM
```

Disk number — File name

Batch-File Processing

Many users will invoke the DOS BACKUP and RESTORE commands from within DOS batch files. To enhance your batch-processing capabilities, BACKUP and RESTORE provide exit status values.

Tip: *The DOS BACKUP command provides the following exit status values.*

Exit Status	Meaning
0	Successful BACKUP operation
1	No files were found to back up
2	Some files were not backed up due to file-sharing conflicts
3	User terminated by CTRL-C
4	Terminated due to an error

Tip: *The DOS RESTORE command provides the following exit status values.*

Exit Status	Meaning
0	Successful RESTORE operation
1	No files were found to restore
3	User terminated by CTRL-C
4	Terminated due to an error

Correcting Disk Fragmentation

In Chapter 4 you learned that throughout the daily operations of adding and deleting files on your disk, it is quite possible for your disk to become fragmented. A *fragmented file* is a file that is dispersed across your disk as shown in Figure 20-7.

As you will recall, the problem with fragmented files is that since they are spread across your disk, it takes DOS longer to access them. The DOS CHKDSK command reports the state of disk fragmentation.

```
C> CHKDSK *.*

Volume DOSDISK    created Jun 17, 1988 3:49a
33439744 bytes total disk space
53248 bytes in 3 hidden files
202752 bytes in 87 directories
23212032 bytes in 2467 user files
14336 bytes in bad sectors
9957376 bytes available on disk

655360 bytes total memory
581968 bytes free

C:\TEST.COM
   Contains 3 non-contiguous blocks.
C:\EGA.CPI
   Contains 7 non-contiguous blocks.
C:\KEYBOARD.SYS
   Contains 4 non-contiguous blocks.
C:\EDLIN.COM
   Contains 4 non-contiguous blocks.
```

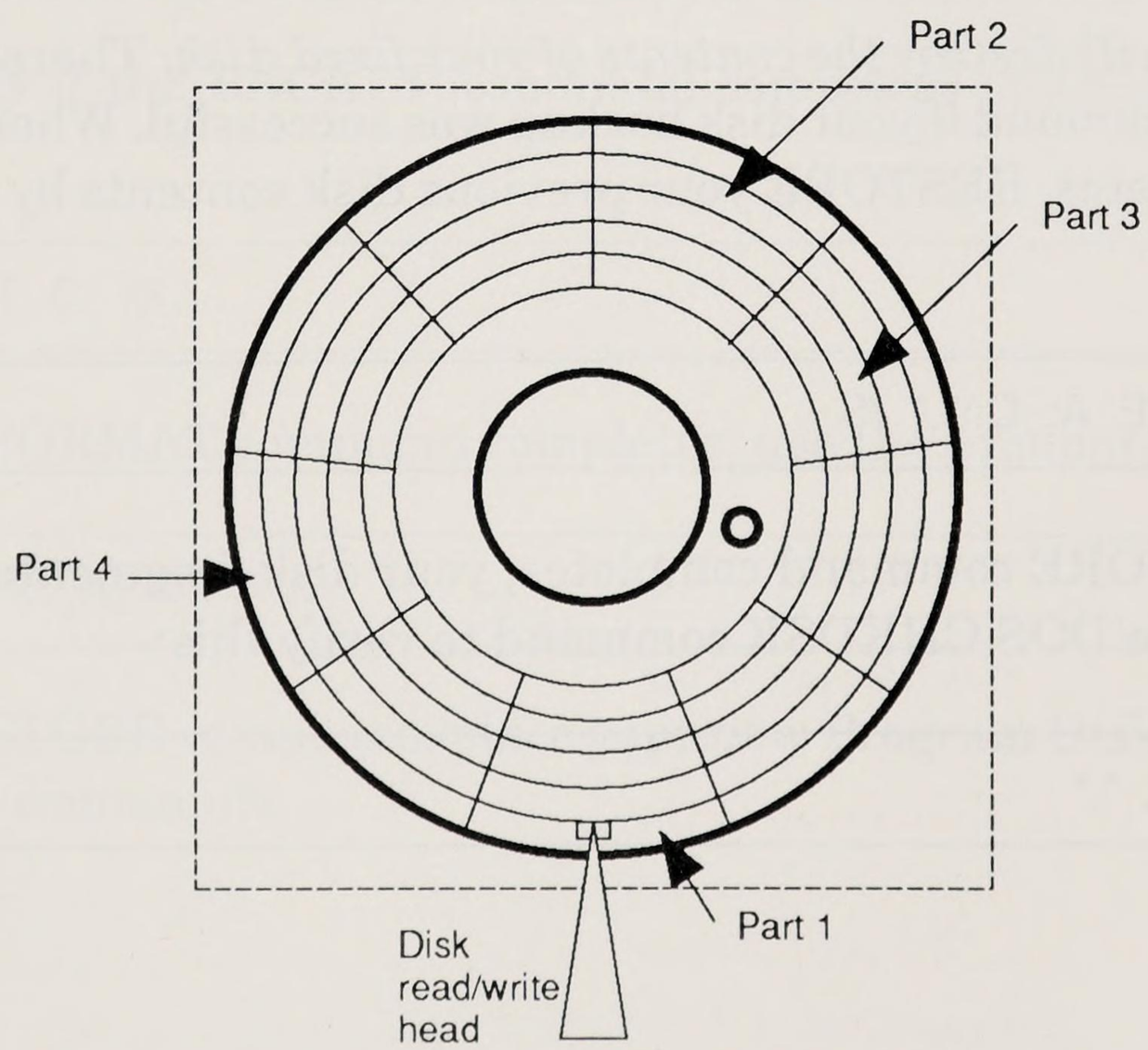


Figure 20-7. A fragmented file

If a large percentage of your files are fragmented, you may want to perform a DOS BACKUP, FORMAT, and RESTORE operation.

Tip: *Once your disk becomes fragmented, the only way that DOS provides to repair it is a BACKUP, FORMAT, and RESTORE operation.*

The first step in this process is to obtain a complete disk backup of all of the files on your disk. Follow the same procedures that you used earlier to determine the number of floppy disks that the backup will require. Next, issue the command

```
C> BACKUP C:\*.* A: /S
```

Only if the BACKUP command completes successfully, place your Program Disk in drive A and issue the command

```
A> FORMAT C:/S
```

This command will destroy the contents of your fixed disk. Therefore, only issue the FORMAT command if your disk backup was successful. When the FORMAT command completes, RESTORE your previous disk contents by using the command

```
A> RESTORE A: C:\*.* /S
```

When the RESTORE command completes, your disk fragmentation should be minimal. Use the DOS CHKDSK command to verify this:

```
C> CHKDSK *.*
```

Upgrading Your Fixed Disk To a New Version of DOS

Earlier, the DOS SYS command and its use in upgrading your fixed disk to a new version of DOS was discussed. At that time, you found that if SYS successfully transferred the hidden DOS system files to your disk, it would display the message

```
System transferred
```

However, if SYS could not upgrade your disk, it would display the message

```
No room for system on destination disk
```

If SYS cannot upgrade your fixed disk to the new version of DOS, you will have to perform a BACKUP, FORMAT, and RESTORE operation to upgrade. To start, boot the new version of DOS and issue the command

```
A> BACKUP C:\*.* A: /S
```

Again, *only* if the BACKUP command completes successfully, issue the command

```
A> FORMAT C: /S
```

Once the FORMAT command completes, use the command

```
A> RESTORE A: C:\*.* /S
```

When RESTORE completes, place your new Program Disk back into drive A and issue the commands

```
A> COPY  COMMAND.COM  C:\
A> COPY  *.*  C:\DOS
```

Repeat this process for your Supplemental Programs Disk:

```
A> COPY  *.*  C:\DOS
```

Your fixed disk is now ready to start DOS.

Tips and Traps Summary

Trap: If you do not back up your disks on a regular basis, you are essentially playing Russian roulette with your files. Eventually, you will lose information or programs.

When to Use BACKUP

Tip: BACKUP allows you to back up specific files, the entire disk, or files meeting specific conditions.

Trap: BACKUP places files on the floppy disk in a unique format that only the DOS BACKUP and RESTORE commands understand.

Getting Started with BACKUP

Trap: Do not use the DOS FORMAT /S command to format your backup disks. In so doing, the system files that DOS places on the disk would consume considerable space that DOS could otherwise utilize.

Tip: Under DOS version 3.3, the BACKUP /F qualifier allows you to use unformatted disks in the target drive. Since doing so increases the amount of time BACKUP requires, always try instead to have sufficient formatted disks available.

Tip: If BACKUP requires more than one disk to perform a backup operation, it will prompt you to insert new disks as required.

Developing a Backup Policy

Tip: Use the CHKDSK command to determine the amount of space that your disk is using.

Tips and Traps Summary (*continued*)

What Is a Log File?

Tip: Under DOS version 3.3, the /L qualifier directs BACKUP to create a log file, which contains a list of the files that DOS has placed on each disk.

Batch-File Processing

Tip: The DOS BACKUP command provides the following exit status values.

Exit Status	Meaning
0	Successful BACKUP operation
1	No files were found to back up
2	Some files were not backed up due to file-sharing conflicts
3	User terminated by CTRL-C
4	Terminated due to an error

Tip: The DOS RESTORE command provides the following exit status values.

Exit Status	Meaning
0	Successful RESTORE operation
1	No files were found to restore
3	User terminated by CTRL-C
4	Terminated due to an error

Tips and Traps Summary (*continued*)

Correcting Disk Fragmentation

Tip: Once your disk becomes fragmented, the only way that DOS provides to repair it is a BACKUP, FORMAT, and RESTORE operation.

21

Using DEBUG to Enhance Your Batch-File Processing

Place your DOS Supplemental Programs Disk in drive A and issue the command

```
A> DIR  DEBUG.COM

Volume in drive A is MS330PP02
Directory of  A:\

DEBUG          COM          15866   7-24-87  12:00a
            1 File(s)  2048 bytes free
```

DEBUG is a programmer's tool that helps programmers find errors (called *bugs*) in programs and then remove (*debug*) them. As you will learn in this chapter, however, DEBUG is very useful to nonprogrammers because it allows you to enhance your batch-processing capabilities. Although DEBUG can at first appear intimidating because it is a programmer's tool, keep in mind that DEBUG

is simply another program. If you follow the examples as they are given in this chapter, DEBUG will be very easy to use. As you will see, by spending a few minutes entering the programs presented here, your batch-processing capabilities will increase considerably.

Tip: *Although DEBUG is a programmer's tool, nonprogrammers can use DEBUG to create simple programs that increase their batch-processing capabilities.*

This chapter examines only a few of DEBUG's capabilities. For more information on DEBUG, refer to the book *DOS: The Complete Reference* (Osborne/McGraw-Hill, 1987).

Creating a Program with DEBUG

Although most of you are not programmers, by following a few simple steps you can create programs using DEBUG. Earlier in this book you found that the DOS PAUSE command temporarily suspends batch-file processing until the user presses a key at the prompt

```
Strike a key when ready . . .
```

However, if the user is not facing the computer screen when this message occurs, it may take a considerable amount of time before the user realizes that DOS is waiting on him or her. In this section, you will create a program called BELL.COM that sounds the computer's bell, which will get the user's attention. Once you create this program, you can place it in the batch file before the PAUSE command, as shown here:

```
BELL
BELL
BELL
PAUSE Turn on your printer
```

To create this program, invoke DEBUG as

```
A> DEBUG  B:BELL.COM
```

DEBUG will respond with

```
A> DEBUG  B:BELL.COM
File not found
-
```

The dash (-) is DEBUG's prompt for you to enter a command. Enter the command **A 100**, which tells DEBUG to start creating your program. DEBUG will display

```
A> DEBUG  B:BELL.COM
File not found
-A 100
15D2:0100
```

Your system may display a set of numbers other than 15D2:0100. These numbers are not important. Type the line **MOV AH,2** and press ENTER, which instructs DOS to output a character:

```
A> DEBUG  B:BELL.COM
File not found
-A 100
15D2:0100  MOV AH,2
```

Next, enter the line **MOV DL,7**, which tells DOS the character that you want to output.

```
A> DEBUG  B:BELL.COM
File not found
-A 100
15D2:0100  MOV AH,2
15d2:0102  MOV DL,7
```

ASCII 7 is the character associated with your computer's bell. When DOS writes this character, your computer will sound its bell. To complete your program, enter the lines

```
A> DEBUG  B:BELL.COM
File not found
-A 100
15D2:0100  MOV AH,2
15D2:0102  MOV DL,7
15D2:0104  INT 21
15D2:0106  MOV AH,4C
15D2:0108  INT 21
15D2:010A
```

Press ENTER and DEBUG will redisplay the prompt:

```
A> DEBUG  B:BELL.COM
File not found
-A 100
15D2:0100 MOV AH,2
15D2:0102 MOV DL,7
15D2:0104 INT 21
15D2:0106 MOV AH,4C
15D2:0108 INT 21
15D2:010A
-
```

Next, you must tell DOS how big the program is that you want to create. To do so, type the line **R CX** and press ENTER, as shown here:

```
A> DEBUG  B:BELL.COM
File not found
-A 100
15D2:0100 MOV AH,2
15D2:0102 MOV DL,7
15D2:0104 INT 21
15D2:0106 MOV AH,4C
15D2:0108 INT 21
15D2:010A
-R CX        (Size of
CX 0000      program)
:
```

Next, type the letter (or number) that corresponds to the last line of the program.

```
A> DEBUG  B:BELL.COM
File not found
-A 100
15D2:0100 MOV AH,2
15D2:0102 MOV DL,7
15D2:0104 INT 21
15D2:0106 MOV AH,4C
15D2:0108 INT 21
15D2:010A
-R CX
CX 0000
:A
-
```

To create the BELL.COM file on disk, type the letter **W** (for the DEBUG write command) and then type **Q** to quit DEBUG:

```
A> DEBUG  B:BELL.COM
File not found
-A 100
15D2:0100 MOV AH,2
15D2:0102 MOV DL,7
15D2:0104 INT 21
15D2:0106 MOV AH,4C
15D2:0108 INT 21
15D2:010A
-R CX
CX 0000
:A
-W
Writing 000A bytes
-Q

A>
```

When you invoke BELL.COM from the DOS prompt or a batch file, your computer's bell will sound. Some explanatory notes are provided to the side of the program, creating

```
A> DEBUG  B:BELL.COM
File not found
-A 100                    ←——— Enter instructions at location 100
15D2:0100 MOV AH,2        ←——— Tell DOS to output a character
15D2:0102 MOV DL,7        ←——— The character to output is 7
15D2:0104 INT 21          ←——— Perform the character output
15D2:0106 MOV AH,4C       ←——— Tell DOS to terminate this program
15D2:0108 INT 21          ←——— Perform the program termination
15D2:010A
-R CX                     ←——— Specify the program size
CX 0000
:A                        ←——— Program size from 010A
-W                        ←——— Write the program to disk
Writing 000A bytes
-Q                        ←——— Exit DEBUG back to DOS

A>
```

As you create more files using DEBUG throughout this chapter, this processing will seem much more straightforward.

Obtaining the Current Disk Drive

Although it is easy for your batch files to set the current disk drive, DOS does not provide a means for determining what the current disk drive is. In this section you will create a program called GETDISK.COM that returns the disk drive number associated with the current disk drive (A = 0, B = 1, C = 2, and so on). When this program completes, it sets the exit status value to the disk drive number. Therefore, you can use the DOS IF ERRORLEVEL command to determine the current disk drive.

Again, invoke DEBUG as

```
A> DEBUG  B:GETDISK.COM
```

At the DEBUG prompt, again enter the line **A 100**. As it turns out, DOS begins its program execution at line 100 (in hexadecimal). That is why you must

use A 100. You want the lines of code that you add to be the first lines DOS executes.

Next, enter the following program:

```
A> DEBUG  B:GETDISK.COM
File not found
-A 100
1641:0100  MOV AH,19
1641:0102  INT 21
1641:0104  MOV AH,4C
1641:0106  INT 21
1641:0108
-R CX
CX 0000
:8
-W
Writing 0008 bytes
-Q

A>
```

This batch file uses GETDISK.COM to display the current disk drive letter:

```
ECHO OFF
GETDISK
IF ERRORLEVEL 0 IF NOT ERRORLEVEL 1 ECHO DRIVE A
IF ERRORLEVEL 1 IF NOT ERRORLEVEL 2 ECHO DRIVE B
IF ERRORLEVEL 2 IF NOT ERRORLEVEL 3 ECHO DRIVE C
```

Determining the State Of Extended CTRL-BREAK Processing

Earlier, this book discussed the fact that the BREAK ON command enables extended CTRL-BREAK checking. The GETBREAK.COM program allows your batch files to determine the current state of extended CTRL-BREAK checking (ON or OFF).

Exit Status	Meaning
0	Extended CTRL-BREAK checking OFF
1	Extended CTRL-BREAK checking ON

To create GETBREAK.COM, use

```
A> DEBUG  B:GETBREAK.COM
File not found
-A 100
1641:0100  MOV AH,33
1641:0102  MOV AL,0
1641:0104  INT 21
1641:0106  MOV AL,DL
1641:0108  MOV AH,4C
1641:010A  INT 21
1641:010C
-R CX
CX 0000
:C
-W
Writing 000C bytes
-Q

A>
```

This batch file determines the state of extended CTRL-BREAK checking and pauses to notify the user:

```
GETBREAK
IF ERRORLEVEL 0 IF NOT ERRORLEVEL 1 PAUSE Extended Checking OFF
IF ERRORLEVEL 1 PAUSE Extended Checking ON
```

Determining the State Of Disk Verification

In a manner very similar to GETBREAK, the GVERIFY.COM program allows your batch files to determine the current state of disk verification (ON or OFF) by using the exit status values shown here:

Exit Status	Meaning
0	Disk verification is OFF
1	Disk verification is ON

To create the GVERIFY.COM program, use DEBUG, as shown here:

```
A> DEBUG  B:GVERIFY.COM
File not found
-A 100
1641:0100  MOV AH,54
1641:0102  INT 21
1641:0104  MOV AH,4C
1641:0106  INT 21
1641:0108
-R CX
CX 0000
:8
-W
Writing 0008 bytes
-Q

A>
```

This batch file displays the current disk verification state to the user.

```
ECHO OFF
GVERIFY
IF ERRORLEVEL 0 IF NOT ERRORLEVEL 1 ECHO VERIFY is OFF
IF ERRORLEVEL 1 ECHO VERIFY is ON
```

Getting a User Response

In many cases, you may wish that your batch files could prompt the user with a message, then obtain a yes or no response such as

```
Do you want to continue (Y/N)?
```

and continue processing depending on the user's response. The GETKEY.COM program returns the key pressed by the user. Again using the IF ERRORLEVEL command, you can determine the user's response, as shown here:

```
ECHO OFF
GETKEY
IF ERRORLEVEL 78 IF NOT ERRORLEVEL 79 ECHO N WAS PRESSED
IF ERRORLEVEL 89 IF NOT ERRORLEVEL 90 ECHO Y WAS PRESSED
```

To create GETKEY.COM, use DEBUG, as shown here:

```
A> DEBUG  B:GETKEY.COM
File not found
-A 100
1641:0100  MOV AH,1
1641:0102  INT 21
1641:0104  MOV AH,4C
1641:0106  INT 21
1641:0108
-R CX
CX 0000
:8
-W
Writing 0008 bytes
-Q

A>
```

Getting a Function-Key Response

Just as the GETKEY.COM program returns a standard key pressed by the user, the GETFKEY.COM program returns the function key pressed, as shown in Figure 21-1. You can create a batch program that displays this information

F1	Shows contents of CONFIG.SYS
F2	Shows contents of AUTOEXEC.BAT
F3	Directory listing of files

This program uses GETFKEY to implement the menu-driven system just displayed.

```
ECHO OFF
CLS
ECHO ———
ECHO | F1    |          Shows contents of CONFIG.SYS
ECHO ———
ECHO OFF
ECHO ———
ECHO | F2    |          Shows contents of AUTOEXEC.BAT
ECHO ———
ECHO OFF
ECHO ———
ECHO | F3    |          Directory listing of files
ECHO ———
:KEY_LOOP
GETFKEY
IF ERRORLEVEL 59 IF NOT ERRORLEVEL 60 TYPE CONFIG.SYS
IF ERRORLEVEL 60 IF NOT ERRORLEVEL 61 TYPE AUTOEXEC.BAT
IF ERRORLEVEL 61 IF NOT ERRORLEVEL 62 DIR
IF ERRORLEVEL 62 GOTO KEY_LOOP
```

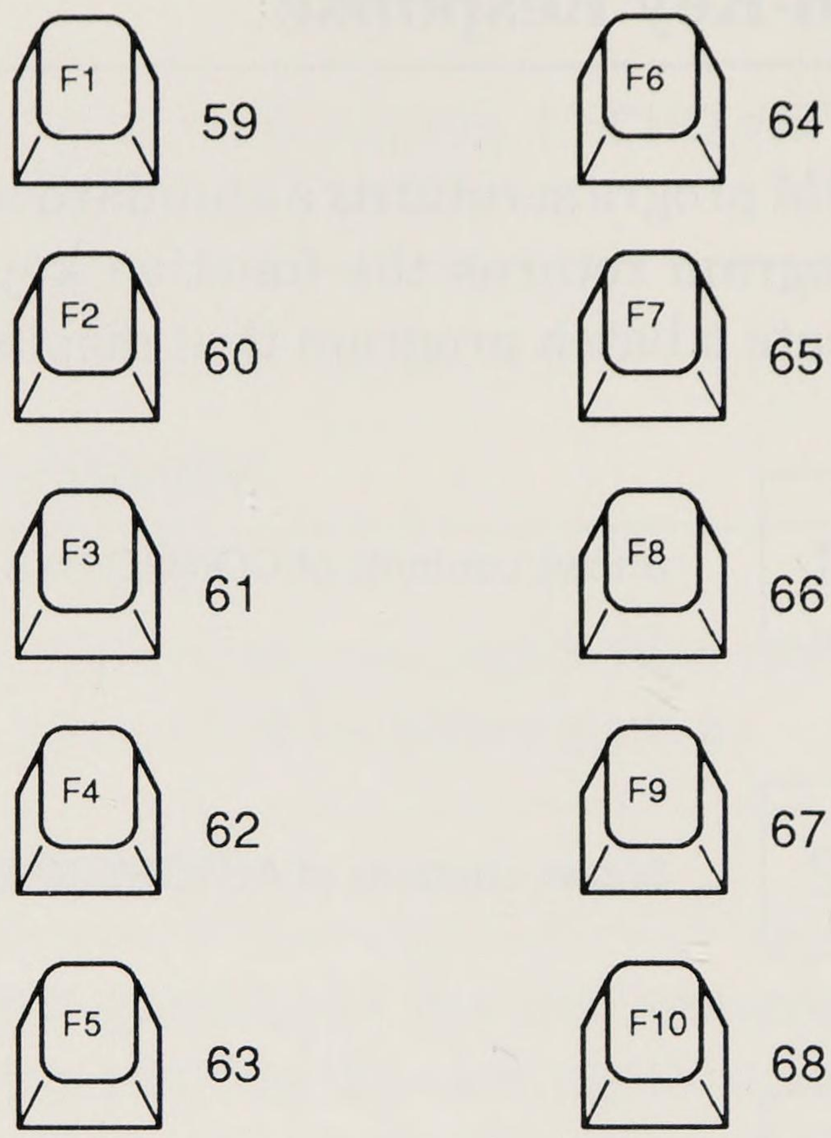


Figure 21-1. GETFKEY.COM returns the function key pressed

This program implements GETFKEY.COM:

```
A> DEBUG  B:GETFKEY.COM
File not found
-A 100
1641:0100  MOV     AH,8
1641:0102  INT 21
1641:0104  CMP     AL,0
1641:0106  JNZ 0100
1641:0108  MOV     AH,8
1641:010A  INT 21
1641:010C  MOV     AH,4C
1641:010E  INT 21
1641:0110
-R CX
CX 0000
:10
-W
Writing 0010 bytes
-Q

A>
```

Admittedly, to the nonprogrammer, several of the concepts in this chapter may seem difficult. However, if you create the command files as shown, you have created several very useful utilities that can increase the functionality of your batch files. Each of the programs that you created in this section took advantage of the DOS *system services.* The book *The DOS Power User's Guide* (Osborne/McGraw-Hill 1987) explains these services in detail.

Tips and Traps Summary

Tip: Although DEBUG is a programmer's tool, nonprogrammers can use DEBUG to create simple programs that increase their batch-processing capabilities.

A

ASCII Codes

Table A-1 lists the ASCII codes for characters.

Table A-1. ASCII Character Codes

DEC	OCTAL	HEX	ASCII	DEC	OCTAL	HEX	ASCII
0	000	00	NUL	7	007	07	BEL
1	001	01	SOH	8	010	08	BS
2	001	02	STX	9	011	09	HT
3	003	03	ETX	10	012	0A	LF
4	004	04	EOT	11	013	0B	VT
5	005	05	ENQ	12	014	0C	FF
6	006	06	ACK	13	015	0D	CR

Table A-1. ASCII Character Codes (continued)

DEC	OCTAL	HEX	ASCII	DEC	OCTAL	HEX	ASCII
14	016	0E	SO	41	050	28	(
15	017	0F	SI	41	051	29	)
16	020	10	DLE	42	052	2A	*
17	021	11	DC1	43	053	2B	+
18	022	12	DC2	44	054	2C	,
19	023	13	DC3	45	055	2D	-
20	024	14	DC4	46	056	2E	.
21	025	15	NAK	47	057	2F	/
22	026	16	SYN	48	060	30	0
23	027	17	ETB	49	061	31	1
24	030	18	CAN	50	062	32	2
25	031	19	EM	51	063	33	3
26	032	1A	SUB	52	064	34	4
27	033	1B	ESC	53	065	35	5
28	034	1C	FS	54	066	36	6
29	035	1D	GS	55	067	37	7
30	036	1E	RS	56	070	38	8
31	037	1F	US	57	071	39	9
32	040	20	SPACE	58	072	3A	:
33	041	21	!	49	073	3B	;
34	042	22	"	60	074	3C	
35	043	23	#	61	075	3D	=
36	044	24	$	62	076	3E	>
37	045	25	%	63	077	3F	?
38	046	26	&	64	100	40	@
39	047	27	’	65	101	41	A

Table A-1. ASCII Character Codes (continued)

DEC	OCTAL	HEX	ASCII	DEC	OCTAL	HEX	ASCII
66	102	42	B	92	134	5C	\
67	103	43	C	93	135	5D	]
68	104	44	D	94	136	5E	^
69	105	45	E	95	137	5F	_
70	106	46	F	96	140	60	‘
71	107	47	G	97	141	61	a
72	110	48	H	98	142	62	b
73	111	49	I	99	143	63	c
74	112	4A	J	100	144	64	d
75	113	4B	K	101	145	65	e
76	114	4C	L	102	146	66	f
77	115	4D	M	103	147	67	g
78	116	4E	N	104	150	68	h
79	117	4F	O	105	151	69	i
80	120	50	P	106	152	A	j
81	121	51	Q	107	153	6B	k
82	122	52	R	108	154	6C	l
83	123	53	S	109	155	6D	m
84	124	54	T	110	156	6E	n
85	125	55	U	111	157	6F	o
86	126	56	V	112	160	70	p
87	127	57	W	113	161	71	q
88	130	58	X	114	162	72	r
89	131	59	Y	115	163	73	s
90	132	5A	Z	116	164	74	t
91	133	5B	[	117	165	75	u

DEC	OCTAL	HEX	ASCII	DEC	OCTAL	HEX	ASCII
118	166	76	v	123	173	7B	{
119	167	77	w	124	174	7C	|
120	170	78	x	125	175	7D	}
121	171	79	y	126	176	7E	~
122	172	7A	z	127	177	7F	DEL

B

Command Reference

This appendix takes a detailed look at each DOS command. As you examine the commands, pay particular attention to the examples that accompany each.

APPEND

FUNCTION Defines the data-file search path that DOS uses each time it fails to locate a file in the current directory or in a specified directory.

COMMAND TYPE External APPEND.EXE

FORMAT

[*drive:*][*path*]APPEND [*d:*][*p*][;[*d:*][*p*]...]

or

[*drive:*][*path*]APPEND [/X][/E] (DOS version 3.3)

or

[*drive:*][path]APPEND ;

where the following is true:

drive: specifies the disk drive containing the APPEND.COM file. If you do not specify a disk drive identification, DOS uses the current default disk drive.

path is the DOS path name of the subdirectory that contains the APPEND.COM file. If you do not specify a DOS path name, DOS uses the current default directory.

d: specifies a disk drive that DOS is to include in the data-file search path.

p specifies a DOS directory to be included in the data-file search path.

... indicates that the disk drive and directories can be specified several times.

/X aids in SEARCH FIRST, FIND FIRST, and EXEC options. This is a DOS version 3.3 qualifier.

/E places an APPEND entry in the DOS environment similar to the PATH entry. Therefore, the DOS SET command can affect APPEND. This is a DOS version 3.3 qualifier.

NOTES Each time DOS cannot find a data file in the specified directory or in the current directory, it checks to see if the user has defined a data-file search path. The DOS APPEND command allows you to define disk drives and directories to be included in the data-file search path.

By default, not all DOS commands use the APPEND data path. If you are using DOS version 3.3, the /X qualifier increases the number of applications that search the data-file search path.

If you invoke APPEND with only a semicolon, DOS removes the data-file search path.

EXAMPLES If DOS cannot find the data file in the current directory, the following command tells DOS to search the root directories on drives C, B, and A, in that order:

```
C> APPEND C:\;B:\;A:\
```

The following APPEND command directs DOS to search \DOS, \UTIL, and then \MISC, in that order:

```
C> APPEND \DOS;\UTIL;\MISC
```

If you are using DOS version 3.3, the /E qualifier directs DOS to place an entry in the environment containing the data-file search path, as in this command:

```
C> APPEND C:\DOS;C:\UTIL /E
```

If you are using DOS version 3.3, you can increase the number of applications that support the data-file search path by using the /X qualifier, as shown here:

```
C> APPEND C:\DOS;C:\UTIL /X
```

If you simply specify a semicolon in the command line, DOS removes the data-file search path, as in this command:

```
C> APPEND ;
```

MESSAGES If APPEND displays the message

```
APPEND / ASSIGN Conflict
```

you have issued the ASSIGN command before issuing the DOS APPEND command. You must remove any disk drive reassignments before invoking APPEND.

If APPEND displays the message

```
APPEND / TopView Conflict
```

you are invoking APPEND while TopView is active. Terminate TopView and reissue the APPEND command.

If APPEND displays the message

```
Incorrect DOS version
```

the executable APPEND.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct APPEND.EXE for the executing version of DOS.

If APPEND displays the message

```
No Append
```

the data-file search path is not currently in use.

If APPEND displays the message

```
APPEND already installed
```

the APPEND command has already been invoked. Once invoked, APPEND installs memory-resident software that handles your data-file search path processing. To modify the data-file search path, omit the disk drive specifier or subdirectory path that precedes APPEND in the command line.

If APPEND displays the message

```
Invalid path
```

one of the subdirectories that you have specified in the data-file search path is invalid. Make sure that you have not mistyped the path name or omitted a semicolon.

If APPEND displays the message

```
Invalid parameter
```

make sure that you are using only the /E or /X qualifier in the command line. Remember, APPEND only supports these qualifiers the first time that you invoke the command.

If APPEND displays the message

```
Invalid path or parameter
```

verify that you have not mistyped a path name or placed a qualifier other than /E or /X in the command line.

If APPEND displays the message

```
Out of environment space
```

you have likely installed a memory-resident program (GRAPHICS, PRINT, or a third-party product) that prevents DOS from expanding the environment to store your new entry. If necessary, restart DOS and issue your APPEND command prior to invoking the memory-resident program. Remember, by using SHELL= in CONFIG.SYS, you can specify the size of your DOS environment. Also, you can invoke APPEND without specifying the /E qualifier.

ASSIGN

FUNCTION Routes disk drive references from one disk drive to another.

COMMAND TYPE External ASSIGN.COM

FORMAT

ASSIGN [*source_drive=target_drive*[...]]

where the following is true:

source_drive is the disk drive identification of the disk from which I/O references are routed.

target_drive is the disk drive identification of the disk to which disk I/O operations will be routed.

... indicates that the command can be repeated several times.

NOTES Many older software packages look on drive A for data or overlay files. If you want to install this kind of software on your fixed disk, you must trick the software into looking on the fixed disk for the files. ASSIGN allows you to do this.

If you invoke ASSIGN without any command-line parameters, ASSIGN restores its original disk drive assignments.

Do not place a colon after each disk drive identification in the ASSIGN command line.

To maintain compatability with future releases of DOS, you should consider using the DOS SUBST command instead of ASSIGN.

Do not use the following commands with an assigned disk:

BACKUP

DISKCOPY

FORMAT
JOIN
LABEL
PRINT
RESTORE
SUBST

EXAMPLES In the following example, DOS disk-I/O operations that reference drive A will be routed to disk C:

```
C> ASSIGN A=C
```

If you invoke a command such as

```
C> DIR A:
```

DOS actually lists the files contained on drive C, as shown here:

```
Volume in drive C has no label
Directory of  C:\MISC

HARDWARE        <DIR>        10-05-87     2:20p
SOFTWARE        <DIR>        10-05-87     2:24p
LEDGER          <DIR>        10-05-87     2:24p
          3 File(s)      1209344 bytes free
```

The following command illustrates that you can perform multiple disk drive assignments on one command line. In this case, DOS will route the disk drive request for either drive A or B to drive C:

```
C> ASSIGN A=C B=C
```

To resume default disk drive assignments, enter ASSIGN at the DOS prompt with no command-line parameters:

```
C> ASSIGN
```

MESSAGES If ASSIGN displays the message

```
Incorrect DOS version
```

the executable ASSIGN.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct ASSIGN.COM for the executing version of DOS.

If ASSIGN displays the message

```
Invalid parameter
```

make sure that your disk drive specifiers are valid, and that you have specified sufficient disk drives. Do not place colons after the disk drive letters.

ATTRIB

FUNCTION Displays or modifies a file's attribute byte.

COMMAND TYPE External ATTRIB.EXE

FORMAT

[*drive:*][*path*]ATTRIB [+A|-A] [+R|-R] *file_specification* [/S]

where the following is true:

drive: specifies the disk drive containing the ATTRIB.EXE file. If you do not specify a disk drive identification, DOS uses the current default disk drive.

path is the DOS path name of the directory that contains the ATTRIB.EXE file. If you do not specify a DOS path name, DOS uses the current default directory.

+A directs ATTRIB to set a file's archive bit.

-A directs ATTRIB to clear a file's archive bit.

+R directs ATTRIB to set a file's read-only bit.

-R directs ATTRIB to clear a file's read-only bit.

file_specification is the complete DOS file specification, including the disk drive and path name of the file(s) to modify. ATTRIB supports DOS wildcard characters.

/S directs DOS to process all files in the directories below the specified directory. This is a DOS version 3.3 qualifier.

NOTES Each DOS file has a directory entry that contains the following fields:

Field	Offset
File name	0
Extension	8
Attribute byte	11
Reserved for DOS	12
Time	22
Date	24
Starting cluster number	26
File size	28

The DOS ATTRIB command modifies a file's attribute byte.

Several DOS commands use a file's attribute to enable selective file processing, such as BACKUP, RESTORE, and XCOPY. By using these commands in conjunction with ATTRIB, you can gain considerable file-processing control.

Get in the habit of using ATTRIB to set your EXE and COM files to read-only, protecting them from errant DOS commands, such as

```
C> DEL *.*
```

EXAMPLES If you do not specify the /A or /R qualifier, as in this command

```
C> ATTRIB *.*
```

ATTRIB displays the current attributes of each file, as shown here:

```
A          C:\DOS\ANSI.EXE
A          C:\DOS\ATTRIB.EXE
A          C:\DOS\FDISK.EXE
A          C:\DOS\GWBASIC.EXE
A          C:\DOS\SUBST.EXE
A          C:\DOS\XCOPY.EXE
```

In the command

```
C> ATTRIB +R CONFIG.SYS
```

ATTRIB will set the CONFIG.SYS file to read-only. In so doing, DOS cannot modify the file's contents. If you attempt to delete a read-only file, for example, DOS displays

```
Access denied
```

The DOS BACKUP /M qualifier directs BACKUP to back up only those files modified since the previous backup. By issuing the command

```
C> ATTRIB +A \*.* /S
```

you can set the archive bit of every file on disk to indicate that a backup is required. The command

```
C> ATTRIB +A \*.* /S
```

marks each file as being backed up.

ATTRIB can be used in conjunction with XCOPY to copy the entire contents of a fixed disk to floppy disks while maintaining the original disk structure. First, mark all the files on disk as requiring a backup, as shown here:

```
C> ATTRIB +A \*.* /S
```

Next, invoke XCOPY, as shown here:

```
C> XCOPY C:\*.* A: /S /M
```

When the floppy disk fills, XCOPY displays a message stating that fact and then terminates. Insert a new floppy disk in the drive and repeat this process. XCOPY will pick up with the first file marked as archived and resume the process at the correct location. Repeat this process until XCOPY finds no files to copy.

The following command sets all of your COM files to read-only, while also setting each to archive required:

```
C> ATTRIB +R +A *.COM
```

If you issue the command

```
C> ATTRIB \*.* /S
```

ATTRIB will display all of the files contained on the disk along with their attributes.

MESSAGES If ATTRIB displays the message

```
Invalid path or file not found
```

either the file or path name specifying the files that ATTRIB is to manipulate is invalid.

If ATTRIB displays the message

```
Invalid drive specifier
```

the disk drive letter that you have specified in the ATTRIB command line before the files to manipulate is invalid. Make sure that you are using an existing disk drive letter (A, B, C, and so on).

If ATTRIB displays the message

```
DOS 2.0 or later required
```

it is telling you that it executes only under DOS 2.0 or later.

If ATTRIB displays the message

```
Invalid number of parameters
```

your command line does not specify a file name for ATTRIB to manipulate.

If ATTRIB displays the message

```
Syntax error
```

your command line contains an option other than +A, -A, +R, or -R.

BACKUP

FUNCTION Backs up one or more files to a new disk.

COMMAND TYPE External BACKUP.COM

FORMAT

[*drive*:][*path*]BACKUP *source*:[*file_spec*] *target*: [/A][/D:*mm-dd-yy*]
[L:*logfile*][/M][/S][T:*hh:mm:ss*][/F]

where the following is true:

drive: specifies the disk drive containing the BACKUP.COM file. If you do not specify a disk drive identification, DOS will use the current default disk drive.

path is the DOS path name of the directory that contains the BACKUP.COM file. If you do not specify a DOS path name, DOS uses the current default directory.

source: specifies the source disk that contains the file(s) to be backed up.

file_spec specifies the DOS path name(s) for the file(s) to back up.

target: specifies the target disk.

/A directs BACKUP to append source files to files on the target disk.

/D:*mm-dd-yy* directs BACKUP to back up files modified since the specified date.

/F tells BACKUP that the target disk is unformatted. This is a DOS version 3.3 qualifier.

/L:*logfile* places an entry for all of the files in the specified log file. BACKUP.LOG is the default file name.

/M directs BACKUP to back up files modified since the last backup.

/S directs BACKUP to back up all subdirectory files.

/T:*hh:mm:ss* directs BACKUP to back up files modified since the specified time. This is a DOS version 3.3 qualifier.

NOTES BACKUP displays the name of each file as the file is successfully backed up.

A 10MB disk requires approximately twenty-five 360K floppy disks or eight 1.2MB disks for a complete backup.

To determine the exact number of disks required to perform a backup, first use the DOS CHKDSK command. Here is an example:

```
C> CHKDSK

Volume DOSDISK created Dec 30, 1988  9:46pm

21309440 bytes total disk space.
0 bytes in 1 hidden files.
```

```
206848 bytes in 90 directories.
13946880 bytes in 1694 user files.
20480 bytes in bad sectors.
7084032 bytes available on disk.

C>
```

The number of disks required is computed by

bytes in use = total disk space - bytes available

Based on the figures in the example, the equation becomes

21309440 - 7084032

= 14225408

Although DOS version 3.3 allows you to use unformatted disks in your backup procedures, this adds a considerable amount of time to your backups. If possible, therefore, always start your backups with sufficient formatted disks.

To support batch processing, BACKUP provides the following exit status codes:

Code	Meaning
0	Successful back up
1	No files found to back up
2	Some files not backed up due to sharing conflicts
3	BACKUP terminated by user with CTRL-C
4	BACKUP terminated by an error

EXAMPLES The following command backs up all of the files on drive C, including those in DOS directories, to the floppy disk in drive A:

```
C> BACKUP C:\*.* A: /S
```

This command uses the BACKUP /A qualifier to add the C:NUMBERS.DAT file to the files contained on the backup disk in drive B:

```
C> BACKUP C:NUMBERS.DAT B: /A
```

This command directs BACKUP to back up only the files created since December 31, 1988:

```
C> BACKUP C:\*.* A: /S /D:12:31:88
```

The following command directs BACKUP to back up all of the files created or modified since the last BACKUP command:

```
C> BACKUP C:\*.* A: /S /A /M
```

Finally, this command backs up all of the files created or modified since the last BACKUP command, writing the list of files to the backup log file BACKUP.LOG. This is a DOS version 3.3 qualifier.

```
C> BACKUP C:\*.* A: /S /A /M /L:BACKUP.LOG
```

MESSAGES If BACKUP displays the message

```
*** Backing up files to drive n: ***
Diskette Number: n
```

it is displaying the disk drive to which it is backing up files, along with the backup disk number that it is creating. You can use the disk number when labeling your backup disks.

If BACKUP displays the message

```
*** Last file not backed up ***
```

it did not back up the last file displayed on the screen. This error normally occurs when the backup target disk becomes full. It occurs only when you are using a fixed disk as a target disk.

If BACKUP displays the message

```
Fixed backup device n is full
```

your target drive for your BACKUP command is a fixed disk. In this case, the disk has become full and BACKUP cannot continue.

If BACKUP displays the message

```
Incorrect DOS version
```

the executable BACKUP.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct BACKUP.COM for the executing version of DOS.

If BACKUP displays the message

```
Insert backup diskette n in drive n:
Strike any key when ready
```

it is prompting you to insert the backup disk number specified in the backup target drive. Once you have done so, press any key for BACKUP to continue.

If BACKUP displays the message

```
Insert last backup diskette in drive n:
Strike any key when ready
```

your BACKUP command line contains the /A qualifier, which directs BACKUP to add the files requiring a backup to the disk in the target drive. In this case, BACKUP is prompting you to place the last disk used in a backup operation into the specified drive. Once you do so, press any key to continue.

If BACKUP displays the message

```
Insufficient memory
```

your system has insufficient RAM for DOS to be able to execute the specified BACKUP command. If you have installed memory-resident software, or a RAM drive, consider restarting DOS without them, so that you can issue the BACKUP command.

If BACKUP displays the message

```
Invalid argument
```

one of the qualifiers in your BACKUP command line is invalid. Review the format of the BACKUP command and reissue the command.

If BACKUP displays the message

```
Invalid drive specification
```

either your source or target disk drive in the BACKUP command line is invalid. Make sure that you have placed a colon after the disk drive letter and that you are using an existing disk drive (A, B, C, and so on). Remember, you cannot use BACKUP when an ASSIGN or SUBST command is in effect.

If BACKUP displays the message

```
Invalid number of parameters
```

your BACKUP command line either contains too many or too few arguments. Review the format of the BACKUP command and reissue the command.

If BACKUP displays the message

```
Invalid path
```

the subdirectory path in your BACKUP command line is invalid. Double-check the spelling of your subdirectory name and ensure that you are specifying a valid path name, starting at the root directory if required.

If BACKUP displays the message

```
Last backup diskette not inserted
Insert last backup diskette in drive n:
Strike any key when ready
```

the disk that you inserted in the BACKUP target drive is not the last disk you used for a BACKUP operation. Insert the correct disk and press any key to continue the backup.

If BACKUP displays the message

```
No such file or directory
```

either the file name or subdirectory path name that you specified as the source of the files to be backed up does not exist. Double-check your spelling and make sure that you are specifying a complete path name, starting at the root directory if required.

If BACKUP displays the message

```
Source and target drives are the same
```

the source and target disk drives that you have specified in your BACKUP command are the same disk drive. BACKUP does not support processing of this type. Reissue the BACKUP command, specifying a different disk drive as the target drive.

If BACKUP displays the message

```
Too many open files
```

it could not continue because DOS was unable to open additional files. Make sure that your CONFIG.SYS file contains the entry FILES=20 and restart DOS. Once DOS restarts, the system will be able to open sufficient files for the BACKUP command to successfully complete.

If BACKUP displays the message

```
Warning! Files in the target drive
n:\root directory will be erased
Strike any key when ready
```

it is warning you that if you continue with the BACKUP operation all of the files in the root directory of the target disk will be erased. Since you should be using newly formatted disks for your backup operations, this error message should be purely informational.

If BACKUP displays the message

```
Warning! Files in the target drive
C:\BACKUP directory will be erased
Strike any key when ready
```

it is warning you that if you continue with the BACKUP operation all of the files in the \BACKUP subdirectory of the specified fixed disk will be deleted. Make sure that you no longer require the files in the directory before continuing.

If BACKUP displays the message

```
Warning! No files were found to back up
```

either it did not find any files requiring a backup, or it did not find any files matching the file specifications in your BACKUP command line.

BREAK

FUNCTION Enables or disables extended CTRL- BREAK checking in DOS.

COMMAND TYPE Internal

FORMAT

BREAK [ON | OFF]

where the following is true:

ON enables extended CTRL-BREAK checking.

OFF disables extended CTRL-BREAK checking.

NOTES By default, DOS checks for a user-entered CTRL-BREAK upon completing keyboard, screen, and printer-I/O operations. If you enable extended CTRL-BREAK checking, DOS also checks for a user-entered CTRL-BREAK upon completing each system service, such as disk read or write operations.

If you invoke BREAK without a command-line parameter, BREAK displays the current state of processing, either ON or OFF. By enabling extended

CTRL-BREAK processing, you increase the system overhead because DOS must check for a CTRL-BREAK upon completing each system service. Therefore, you may want to enable this checking during program development. Most users, however, will leave BREAK set at OFF.

By default, DOS boots with BREAK set at OFF. If you want extended CTRL-BREAK processing enabled at system startup, use the BREAK=ON entry in CONFIG.SYS.

EXAMPLES This command enables DOS extended CTRL-BREAK checking:

```
C> BREAK ON
```

As stated, if you invoke BREAK without a command-line parameter, as shown here

```
C> BREAK
BREAK is on
```

BREAK displays its current state of extended CTRL-BREAK checking, either ON or OFF.

MESSAGES If BREAK displays the message

```
Must specify ON or OFF
```

you have invoked the BREAK command specifying a command-line parameter other than ON or OFF. Reissue the command with either no command-line parameters (to display the current break setting) or the word ON (to enable extended checking) or OFF (to disable extended checking).

If you simply invoke BREAK without any command-line arguments, BREAK will display the current state of extended CTRL-BREAK checking (either ON or OFF). Otherwise, you must specify either ON or OFF to enable or disable extended CTRL-BREAK checking, respectively.

CALL

FUNCTION Invokes a nested batch procedure from within a batch file.

COMMAND TYPE Internal

FORMAT

CALL bat_file [parameters]

where the following is true:

bat_file specifies the name of the DOS batch file to invoke. CALL allows you to use a complete file specification, including a subdirectory path and disk drive identifier.

parameters are optional parameters for the nested batch file.

NOTES CALL is a DOS version 3.3 command that allows you to invoke a batch file from within a second batch file.

By default, if one DOS batch procedure invokes a second, or nested, batch file without the use of CALL or COMMAND/C, DOS completes batch-file processing when the nested batch file completes, regardless of the commands that may remain in this batch file. By placing the CALL command before the batch-file name, DOS will execute all of the commands present in both batch files.

EXAMPLE The following batch file invokes the DOS DATE command, followed by a nested batch file called BATFILE, and then issues the DOS TIME command.

```
DATE
CALL BATFILE
TIME
```

Since the batch file uses the CALL command, DOS will return control to the original batch file so that the TIME command can execute.

CHCP

FUNCTION Displays or changes the current codepage.

COMMAND TYPE Internal

FORMAT

CHCP [*code_ page*]

where *code_ page* specifies the desired codepage, which must have been previously prepared by the system as either the primary or secondary codepage in CONFIG.SYS.

NOTES Each time DOS displays a character on the screen, it must first map the ASCII value of the character to a specific letter in a character set. DOS uses codepages to map the letters to specific characters.

DOS allows you to use a different character set for your DOS session, offering you international support. To select an alternative codepage, you must previously issue the NLSFUNC command.

Valid codepage entries include

437	United States
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

If you invoke CHCP without a command-line parameter, CHCP displays the current codepage. This is a DOS version 3.3 command.

EXAMPLES The following command directs CHCP to select the Nordic codepage. Remember that this codepage must have previously been prepared by a NLSFUNC command:

```
C> CHCP 865
```

To display the current codepage, type **CHCP** without any command-line parameters:

```
C> CHCP
```

MESSAGES The following error messages are common to this command. Use this list of messages to resolve errors you experience when you issue commands from the DOS prompt.

If CHCP displays the message

```
Code page nnnn not prepared for system
```

make sure that you have installed the PRINTER.SYS and DISPLAY.SYS device drivers in CONFIG.SYS as required. Next, make sure that you have issued the NLSFUNC command and prepared the codepage for use by the MODE command.

If CHCP displays the message

```
File not found
```

it could not find the COUNTRY.SYS file. Make sure that the COUNTRY.SYS file resides in the current directory, or that you specify the location of this file when you invoke NLSFUNC.

CHDIR

FUNCTION Changes or displays the default directory.

COMMAND TYPE Internal

FORMAT

CHDIR [*drive:*][*path*]

or

CD [*drive:*][*path*]

where the following is true:

drive: specifies the disk drive that contains the directory you want as the default. If you omit the disk drive, CHDIR uses the current default disk drive.

path specifies the DOS path name of the directory that you want to be the current directory. If you omit the path, CHDIR displays the current directory.

NOTES CHDIR changes or displays the current directory name for the specified disk drive. If you do not specify a drive, CHDIR uses the current default drive. If you do not specify a path name, CHDIR displays the current default directory name.

CHDIR performs the following processing each time you specify a DOS path name: If the path name is preceded with a backslash (\) DOS begins its search for the directory at the root. If the path name does not begin with a backslash, the directory must reside below the current default directory.

DOS predefines two directory names, as shown here:

.. parent directory of the current directory

. current default directory

Verify this by performing a listing of a DOS directory, as shown here:

```
Volume in drive C is DOSDISK
Directory of C:\SUBDIR

.          <DIR>      11-01-87    11:34a
..         <DIR>      11-01-87    11:34a

        2 File(s)     7182336 bytes free
```

If you issue the following directory command, DOS displays a directory listing of the current directory:

```
C> DIR .

Volume in drive C is DOSDISK
Directory of C:\SUBDIR

.          <DIR>      11-01-87    11:34a
..         <DIR>      11-01-87    11:34a

        2 File(s)     7182336 bytes free
```

However, if you use

```
C> DIR ..
```

DOS displays the files in the parent directory.

EXAMPLES As stated, if you invoke CHDIR without a path name, CHDIR displays the current directory, as shown here:

```
C> CD
C:\SUBDIR
```

This command is valid with a disk drive specifier, as shown here:

```
C> CHDIR B:
B:\UTIL
```

The following DOS CHDIR command selects the directory DOS\BATCH on drive C as the current default directory:

```
C> CHDIR C:\DOS\BATCH
```

You could issue this command as two separate commands, first selecting the DOS directory

```
C> CD C:\DOS
```

and then selecting BATCH:

```
C> CD C:BATCH
```

Notice the use of the leading backslash in the first command:

```
C> CD C:\DOS
```

When the backslash is used, DOS looks for the DOS directory in the root directory of drive C. In the second command, however,

```
C> CD C:BATCH
```

CHDIR does not look for the directory in the root, but rather in a directory below the current directory (which is the DOS directory).

If you had instead issued the command

```
C> CHDIR C:\BATCH
```

CHDIR would not have found the directory since it would have looked in the root instead of the current directory.

MESSAGES If CHDIR displays the message

```
Invalid directory
```

the subdirectory (or a path in the directory) that you are specifying in the command does not exist. Recheck the spelling of the path, ensuring that it contains backslashes as required.

If CHDIR displays the message

```
Invalid drive specification
```

the disk drive letter that you have specified in the CHDIR command line is invalid. Make sure that you have placed a colon after the disk drive letter and that you are using an existing disk drive (A, B, C, and so on).

CHKDSK

FUNCTION Checks a disk's current status.

COMMAND TYPE External CHKDSK.COM

FORMAT

[*drive:*][*path*]CHKDSK [*d:*][*p*][*filename*] [/F] [/V]

where the following is true:

drive: specifies the disk drive containing the CHKDSK.COM file. If you do not specify a disk drive identification, DOS uses the current default disk drive.

path is the DOS path name of the directory that contains the CHKDSK.COM file. If you do not specify a DOS path name, DOS uses the current default directory.

d: is the disk drive that CHKDSK is to examine.

p specifies a DOS directory containing files that CHKDSK is to examine for disk fragmentation.

filename is the file name and extension of the file (or files) that CHKDSK is to examine for disk fragmentation.

/F directs CHKDSK to fix errors found in a directory or file allocation table (FAT).

/V directs CHKDSK to display the names of all files on the disk.

NOTES CHKDSK reports on the status of the following disk conditions:

- The amount of free, used, and corrupted disk space
- The number of hidden files
- The amount of free and used memory

Periodically, because day-to-day disk operations cause wear and tear on the storage media, files become corrupted and lose sectors. CHKDSK allows you to view and repair such discrepancies.

During the everyday use of a disk, DOS files can also become fragmented, which means their contents are dispersed throughout the entire disk. Fragmented files increase system overhead on file-I/O operations because the disk drive must rotate the disk several additional times in order to read the disk. The DOS CHKDSK command displays information on fragmented files. Once your disk becomes severely fragmented, you should consider a system backup-and-restore operation.

CHKDSK does not work in conjunction with JOIN or SUBST.

If you operate a local-area network, do not try to repair your disk. When DOS has files open (as may be the case with a network) it cannot correctly update the FAT.

By default, CHKDSK will only report disk errors, it will not attempt to fix them. To write the actual corrections to disk, you must use the /F qualifier.

EXAMPLES This command displays the state of the current disk:

```
C> CHKDSK
```

An example output might be

```
Volume DOSDISK created Oct 30, 1988  9:46pm

21309440 bytes total disk space.
0 bytes in 1 hidden files.
208896 bytes in 91 directories.
13871104 bytes in 1714 user files.
20480 bytes in bad sectors.
7206912 bytes available on disk.

524288 bytes total storage
401184 bytes free
```

If you specify a file or use DOS wildcard characters, CHKDSK reports on disk fragmentation:

```
C> CHKDSK *.*
```

If fragmented files exist, CHKDSK displays

```
C:\FILENAME.EXT
contains n non-contiguous blocks.
```

If CHKDSK discovers errors while examining your disk's contents, it displays

```
Volume DOSDISK created Oct 30, 1988  9:46pm

Errors found. F parameter not specified.
Corrections will not be written to disk.

1 lost clusters found in 1 chains.
Convert lost chains to files (Y/N)?
2048 bytes disk space would be freed.
```

To direct CHKDSK to repair the error, use the /F qualifier

```
C> CHKDSK /F
```

You can also use CHKDSK to locate a file quickly. If you invoke CHKDSK with the /V qualifier, CHKDSK displays the name of every file on the disk:

```
C> CHKDSK /V
```

MESSAGES If CHKDSK displays the message

```
Directory is joined
```

it cannot process the specified subdirectory because the directory is joined to another DOS disk. CHKDSK cannot process a subdirectory that is joined.

If CHKDSK displays the message

```
Directory is totally empty, no . or ..
```

the specified subdirectory is probably damaged. In most cases, this error message is an indication of a severely damaged disk.

If CHKDSK displays the message

```
Disk error reading (writing) FAT n
```

it cannot read (or write) the specified file allocation table (FAT). DOS tracks its files by recording which sectors are associated with a file in a table called the file allocation table, which DOS stores on disk. So that a single disk error does not prevent DOS from accessing your files, DOS provides two identical file allocation tables. If one table becomes corrupted, DOS uses the second table.

If CHKDSK displays the message

```
Errors found, F parameter not specified
Corrections will not be written to disk
```

it found errors while examining the specified disk. Since you did not place the /F qualifier in the CHKDSK command line, CHKDSK will not record its corrections to disk. In most cases, you will want to reissue the CHKDSK command including the /F qualifier.

If CHKDSK displays the message

```
File allocation table bad drive n:
```

it cannot access the file allocation table on the specified disk drive. This error message is associated with severe disk errors.

If CHKDSK displays the message

```
File not found
```

it could not find the specified file.

If CHKDSK displays the message

```
First cluster number is invalid, entry truncated
```

the specified file contains an invalid cluster number as the first entry in the file allocation table. CHKDSK has truncated the file to zero bytes.

If CHKDSK displays the message

```
General Failure error reading drive n:
```

the disk in the specified drive has probably not yet been formatted, or is severely damaged.

If CHKDSK displays the message

```
Has invalid cluster, file truncated
```

it has found an invalid cluster entry in the file allocation table for the specified file. As a result, CHKDSK has truncated the size of the file to the last correct entry.

If CHKDSK displays the message

```
Incorrect DOS version
```

the executable CHKDSK.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct CHKDSK.COM for the executing version of DOS.

If CHKDSK displays the message

```
Insufficient memory
```

your system has insufficient RAM for DOS to be able to execute the CHKDSK command. If you have installed memory-resident software or a RAM drive, you might consider restarting DOS without them so that you can issue the CHKDSK command.

If CHKDSK displays the message

```
Insufficient space in root directory
```

it has been creating files in the root directory from the lost clusters it has found. These files are named in the form FILE*nnnn*.CHK. In this case, your root directory is full. Copy the files to a new disk, delete them from this disk, and repeat the CHKDSK command.

If CHKDSK displays the message

```
Invalid current directory
Processing cannot continue
```

your disk is severely damaged. Save as many of the files as you can from the disk drive by copying them to a new location.

If CHKDSK displays the message

```
Invalid drive specification
```

the disk drive you specified in your CHKDSK command is invalid. Make sure that you have placed a colon after the disk drive letter and that you are using an existing disk drive (A, B, C, and so on).

If CHKDSK displays the message

```
Path not found
```

it could not locate the path specified in the command line that it is to examine for disk contiguity. Double-check your spelling and directory name, and reissue the CHKDSK command.

If CHKDSK displays the message

```
Probable non-DOS disk
Continue (Y/N)?
```

it is telling you that it cannot identify the format used on the specified disk and it needs to know whether you want it to continue processing. If you type **Y** and press ENTER, CHKDSK will continue. To terminate processing, type **N** and press ENTER.

If CHKDSK displays the message

```
Unrecoverable error in directory
Convert directory to a file (Y/N)?
```

it has encountered an error in the specified directory that prevents it from continuing its processing. CHKDSK wants to know whether it should convert the damaged directory to a file. In some cases, you may want to delete the damaged directory from your disk. If so, type **Y** and press ENTER, otherwise, type **N**.

If CHKDSK displays the message

```
. Does not exist
```

the abbreviated name for the current directory does not exist. If this error occurs, you should consider making a complete backup of all the files on your disk. This may be a sign of disk errors to come.

If CHKDSK displays the message

```
.. Does not exist
```

the abbreviated name for the parent directory does not exist. If this error occurs, you should consider making a complete backup of all the files on your disk. This may be a sign of disk errors to come.

If CHKDSK displays the message

```
FILENAME.EXT is crossed linked on cluster n
```

DOS has erroneously assigned two files to point to the same cluster (set of sectors) on disk. Copy both of the files to a new disk and delete them. One of the files will be damaged (with missing information). If the file is a text file, you will have to replace the missing data. If the file contains a program, do not execute it. Executing a damaged program can damage your disks and the information that they contain.

If CHKDSK displays the message

```
n lost clusters found in n chains
Convert lost chains to files (Y/N)?
```

your disk has pointers to files in the file allocation table that no longer exist. CHKDSK is prompting you to decide whether to convert the lost pointers to files in the root directory that have the form FILE*nnnn*.CHK, which you can examine later for lost information. If your CHKDSK command line did not contain the /F qualifier, you will have to reinvoke CHKDSK specifying /F for the lost pointers to be written to disk.

If CHKDSK displays the message

```
Allocation error, size adjusted
```

the size of the specified file in a directory listing is inconsistent with the number of bytes that CHKDSK found in the file allocation table. CHKDSK has corrected the error.

If CHKDSK displays the message

```
All specified file(s) are contiguous
```

your CHKDSK command line included a file specification that CHKDSK is to examine for disk contiguity. In this case, all of the files were contiguous.

If CHKDSK displays the message

```
Cannot CHDIR to pathname
tree past this point not processed
```

it cannot continue processing your subdirectory due to disk errors. Copy as many files as you can from your disk to backup disks. This error message is associated with a damaged disk.

If CHKDSK displays the message

```
Cannot CHDIR to root
```

it cannot select the root directory as the current directory due to a disk error. Copy as many files as you can from your disk to backup disks. This error message is associated with a damaged disk.

If CHKDSK displays the message

```
Cannot CHKDSK a Network drive
```

the disk drive that you have specified in the CHKDSK command line is a network drive. DOS does not allow you to use CHKDSK on a network device.

If CHKDSK displays the message

```
Cannot CHKDSK a SUBSTed or ASSIGNed drive
```

the disk drive that you have specified in the CHKDSK command line has been used previously in either an ASSIGN or SUBST command. DOS will not allow you to use CHKDSK on a "pretender" device.

If CHKDSK displays the message

```
CHDIR .. failed, trying alternate method
```

it could not select the parent directory as the current directory by using the .. abbreviation. CHKDSK is trying an alternative method of selecting the current directory. If your disk contains serious errors, CHKDSK will not be able to continue processing.

If CHKDSK displays the message

```
FILENAME.EXT contains n contiguous blocks
```

your CHKDSK command line included a file specification that CHKDSK is to examine for disk contiguity. In this case, CHKDSK is displaying the number of clusters in a file that are not contiguous. If many of your files are noncontiguous, you may consider a BACKUP, FORMAT, and RESTORE operation.

CLS

FUNCTION Clears the screen display.

COMMAND TYPE Internal

FORMAT

```
CLS
```

NOTES CLS does not affect video attributes.
CLS places the cursor and DOS prompt in the home (upper-left) position.

EXAMPLE In the following example, DOS erases the current screen contents, placing the cursor and the current DOS prompt in the home position:

```
C> CLS
```

COMMAND

FUNCTION Loads a secondary command processor.

COMMAND TYPE External COMMAND.COM

FORMAT

COMMAND [*drive:*][*path*] [/C *string*][/E:*num_bytes*][/P]

where the following is true:

drive: specifies the disk drive identification of the disk containing the secondary command processor. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the DOS directory that contains the command processor. If you do not specify a path name, DOS uses the current default directory.

/C *string* directs DOS to execute the command specified by *string*. You can use this parameter for nested batch-file invocations.

/E:*num_bytes* specifies the space DOS must allocate for the secondary command processor environment. *Num_bytes* must be between 160 and 32,767 bytes. The default value is 160.

/P directs DOS to leave the secondary command processor permanently in memory.

NOTES Each time you load a secondary command processor, it obtains its own copy of the DOS environment space.

Most users use COMMAND to invoke nested batch procedures, as shown here:

```
CLS
COMMAND /C BATFILE
DATE
```

If you use the /P and /C parameters together, COMMAND ignores /P.

To terminate a secondary command processor, use the DOS EXIT command.

EXAMPLES As stated, most users use COMMAND to invoke a nested batch procedure, as shown here:

```
DATE
COMMAND /C BATFILE
TIME
```

When DOS completes the DATE command, it will invoke the specified batch file, execute the commands that it contains, and then return control to this batch file to execute the DOS TIME command. If you had instead invoked the batch file without COMMAND /C, DOS would never return control to the original batch file, terminating processing at the end of the nested batch file.

In this case, COMMAND allows the user to issue DOS commands during the execution of a batch file:

```
DATE
COMMAND
TIME
```

When DOS executes COMMAND, it will display the DOS prompt, allowing the user to issue DOS commands. Once the user is done entering DOS commands, he or she types EXIT and DOS will return control to the batch file, executing the DOS TIME command.

MESSAGES If COMMAND displays the message

```
Bad or missing Command Interpreter
```

DOS could not locate the command interpreter COMMAND.COM in the root directory or directory specified in the SHELL= entry in CONFIG.SYS.

If COMMAND displays the message

```
Specified COMMAND search directory bad
```

the subdirectory path specified as a parameter to COMMAND is invalid. Use the EXIT command to terminate the command processor and reissue COMMAND with the correct directory path.

If COMMAND displays the message

```
Invalid environment size specified
```

your COMMAND command line contains the /E qualifier, which allows you to specify an environment size from 160 to 32,767. In this case, the value that you have specified is invalid.

COMP

FUNCTION Displays the first ten differences between two files.

COMMAND TYPE External COMP.COM

FORMAT

[*drive:*][*path*]COMP *file_spec1 file_spec2*

where the following is true:

drive: specifies the disk drive identification of the disk containing the COMP.COM file. If you do not specify a disk, DOS uses the current default disk drive.

path specifies the name of the directory that contains the COMP.COM file. If you do not specify a directory, DOS uses the current default directory.

file_spec1 and *file_spec2* are the complete DOS path names of the files to compare. COMP supports DOS wildcard characters.

NOTES COMP displays the differences as hexadecimal offsets into the file.

If the files are identical, COMP displays

```
Files compare OK
```

If COMP locates more than ten differences, it displays

```
10 Mismatches - ending compare
```

COMP does not compare files of different sizes.

EXAMPLES In the following example, COMP compares the contents of the A.DAT file to B.DAT:

```
C> COMP A.DAT B.DAT
```

Assuming that each file contains

```
C> TYPE A.DAT
A
AA

C> TYPE B.DAT
B
BB
```

the command displays

```
Compare file C:A.DAT and file C:B.DAT

COMPARE error at OFFSET 0
File 1 = 41
File 2 = 42

Compare error at OFFSET 3
File 1 = 41
File 2 = 42
```

```
Compare error at OFFSET 4
File 1 = 41
File 2 = 42

Do you want to compare more files (Y/N)?
```

If you omit the file name from the secondary file, DOS matches the file name on the specified drive with the primary file:

```
C> COMP A.DAT B:
```

If you do not specify files in the COMP command line, COMP prompts you for them, as shown here:

```
C> COMP
Enter the first filename.
A.DAT

Enter the second filename.
B.DAT
```

MESSAGES The following error messages are common to this command. Use this list of messages to resolve errors you experience when you issue commands from the DOS prompt.

If COMP displays the message

```
Compare error at offset nnnnnnnn File1=nn File2=nn
```

COMP has located a difference between the two files and is displaying the offset location within the files of the difference, along with the byte values that differ. Remember, COMP displays these values in hexadecimal.

If COMP displays the message

```
EOF mark not found
```

the last character in one of the files examined was not an end-of-file character ^Z. DOS uses the CTRL-Z character as the end of file marker for most text files. In most cases, this message is purely informational.

If COMP displays the message

```
FILENAME.EXT - File not found
```

COMP could not locate the file specified in the command line. Double-check your spelling and current directory, reissuing the COMP command as required.

If COMP displays the message

```
PATHNAME Invalid path
```

the subdirectory specified in the COMP command line does not exist. Reissue the COMP command with the correct directory name.

If COMP displays the message

```
Files are different sizes
```

the two files specified in the COMP command line differ in size. COMP will not compare files of different size. If you are using DOS version 3.3, use the FC command.

If COMP displays the message

```
Invalid drive specification
```

one of the file names specified in the COMP command line is using an invalid disk drive letter. Make sure that you have placed a colon after the disk drive letter and that you are using a valid disk drive letter (A, B, C, and so on).

If COMP displays the message

```
10 Mismatches - ending compare
```

it has identified ten differences between the two specified files and is terminating the comparision. If you are using DOS version 3.3, the FC command allows you to display all of the differences between two files.

If COMP displays the message

```
Access denied
```

one of the files specified in the COMP command line is locked by another network program that prevents COMP from accessing the file. Wait a few minutes and reissue the command.

If COMP displays the message

```
Cannot compare a file to itself
```

the primary and secondary file names in the COMP command line are the same. COMP will not compare one file to itself.

If COMP displays the message

```
Enter primary file name
```

your COMP command line did not specify the files that COMP is to compare. As such, COMP is prompting you for them. In this case, simply type in the desired file name and press ENTER.

If COMP displays the message

```
Enter 2nd file name or drive id
```

your COMP command line did not specify the files that COMP is to compare. As such, COMP is prompting you for them. In this case, simply type in the desired file name or disk drive and press ENTER. If you simply type in a disk drive letter, COMP will search for a file on that disk drive that matches the primary file name.

If COMP displays the message

```
Compare more files (Y/N)?
```

it has completed one file comparison and is prompting you to specify whether or not you want to perform another file comparison. If so, type **Y** and press ENTER; COMP will prompt you for the files to compare. If you instead type **N**, COMP will terminate and return control to DOS.

If COMP displays the message

```
Incorrect DOS version
```

the executable COMP.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct COMP.COM for the executing version of DOS.

If COMP displays the message

```
Insufficient memory
```

your system does not have enough RAM for DOS to be able to execute the COMP command. If you have installed memory-resident software or a RAM drive, consider restarting DOS without them so that you can issue the COMP command.

If COMP displays the message

```
Too many files open
```

COMP was unable to open the files to compare due to the fact that DOS already has too many files open. If this error occurs, make sure that your CONFIG.SYS contains, at the minimum, an entry of FILES=20. Once you restart DOS with this entry, COMP will be able to complete.

If COMP displays the message

```
File not found
```

it could not locate one of the files specified in the COMP command line. Double-check your spelling and current directory, reissuing the command as required.

If COMP displays the message

```
Invalid path
```

either the primary or secondary file specified in the COMP command line contains a subdirectory path name that does not exist. Recheck your spelling and subdirectory specifications reissuing the COMP command as required.

If COMP displays the message

```
File sharing conflict
```

it could not access one of the files specified in the COMP command line do to network file sharing. Wait a few minutes for the file to be released by the program that is using it and reissue the COMP command.

COPY

FUNCTION Copies one or more files to a new destination.

COMMAND TYPE Internal

FORMAT

COPY *source_file target_file* [/V][/A][/B]

or

COPY *source1+source2* [...] *target_file* [/V][/A][/B]

where the following is true:

source_file specifies the complete DOS file specification of the file to be copied.

target_file is the name of the destination file. If a file matching the name of *target_file* exists, COPY overwrites it.

/V tells COPY to use disk verification to ensure that a successful copy occurred. This qualifier adds processing overhead; however, it prevents a hardware error from rendering the contents of the source and target files inconsistent.

/A informs COPY that the preceding file was an ASCII file.

/B informs COPY that the preceding file was a binary file.

NOTES COPY fully supports DOS wildcard characters.

To combine multiple files into one file, use the plus sign (+) between the source files desired.

COPY will not allow you to copy a file to itself. If you attempt to do so, COPY displays

```
File cannot be copied onto itself
        0 file(s) copied.
```

WARNING: COPY overwrites target files with the same name.

EXAMPLES The following command copies the contents of the CONFIG.SYS file to a file on drive B with the same name:

```
C> COPY A:CONFIG.SYS B:CONFIG.SYS
```

It is identical in function to

```
C> COPY A:CONFIG.SYS B:
```

or

```
C> B:
[B:\] COPY A:CONFIG.SYS
```

The following command copies all of the files on drive A to drive B:

```
C> COPY A:*.* B:*.*
```

If you want to copy files contained in directories below the current directory, use the XCOPY command.

This command uses the plus sign to append the files B and C to the file A, thus creating a file called D:

```
C> COPY A+B+C D
```

The following command allows you to append text from the keyboard to an existing file, creating a new file:

```
C> COPY FILENAME+CON: NEWFILE.EXT
```

MESSAGES The following error messages are common to this command. Use this list of messages to resolve errors you experience when you issue commands from the DOS prompt.

If COPY displays the message

```
Cannot do binary reads from a device
```

you have placed the /B qualifier that directs COPY to perform a binary copy after a device name. When COPY performs a binary copy, COPY does not use the CTRL-Z (^Z) character to indicate an end of file. Since CTRL-Z is the only means of telling COPY when you have entered your required information from a device, COPY cannot perform a binary copy from a file.

If COPY displays the message

```
Contents of destination lost before copy
```

the destination file name is also a file specified as a source file in the COPY operation. In this command, COPY became confused during its processing and has overwritten the file contents. If you are using the destination file as a source file in this manner, try to place the file name first in the COPY command or at least make sure that a file name follows the specified file.

If COPY displays the message

```
File cannot be copied onto itself
```

the source and destination for a file copy operation are the same. In most cases, this error occurs when you have omitted a disk drive specifier or subdirectory in the target file name.

If COPY displays the message

```
Invalid directory
```

either the source or target directory in the COPY command line does not exist. Repeat the command with the correct directory name.

If COPY displays the message

```
FILENAME.EXT File not found
```

COPY could not locate the source file as specified. Double-check your spelling and the current directory and drive and repeat the command with the correct source file name.

CTTY

FUNCTION Selects a new device from which commands are issued.

COMMAND TYPE Internal

FORMAT

CTTY *device_name*

where *device_name* specifies the device from which commands are received (either CON or AUX).

NOTES Most users will never issue this command.

If you have a terminal attached to your serial port, you can direct DOS to obtain its commands from that port by entering the command

```
A> CTTY AUX
```

To resume normal operations, you must enter the following command from the terminal device:

```
A> CTTY CON
```

MESSAGES If CTTY displays the message

```
Invalid device
```

your command line specifies either an invalid device name or no device name. Only invoke CTTY when your system has a terminal device attached to a serial port.

DATE

FUNCTION Sets the DOS system date.

COMMAND TYPE Internal

FORMAT

DATE [*mm-dd-yy*]

or

DATE [*dd-mm-yy*]

or

DATE [*yy-mm-dd*]

where the following is true:

mm is the desired month (1-12).

dd is the desired day (1-31).

yy is the desired year (80-99). DATE also allows you to include the century in the form 19*yy* or 20*yy*.

NOTES The DATE format that you use depends on the COUNTRY= specifier in CONFIG.SYS. If you do not specify a date, DATE displays the current date.

In DOS versions prior to 3.3, DATE does not modify the AT system clock. Users of these earlier versions must use the Setup disk provided with the *Guide to Operations* to change the AT system clock. DOS version 3.3 TIME and DATE commands set this clock.

The date is an optional command-line parameter. If you omit the date, DATE prompts you for it.

EXAMPLES In the following example, since the date was not present on the DOS command line, DATE prompted the user for it:

```
C> DATE

The current date is: Sun 11-21-1988
Enter the new date: (mm-dd-yy)
```

If you want to display the system date without modifying it, press ENTER at the date prompt. DATE leaves the system date unchanged.

In this example, DATE sets the system date to 8 December 1988:

```
C> DATE 12/08/88
```

This command is identical to

```
C> DATE 12/08/1988
```

DATE fully supports a four-digit year. Use this command to set a date from the DATE prompt:

```
C> DATE

The current date is: Sun 11-01-1988
Enter the new date: (mm-dd-yy) 12-08-88
```

MESSAGES If DATE displays the message

```
Invalid date
Enter new date (mm-dd-yy):
```

make sure that you are entering the current system date with the month preceding the day as shown in the format given. Retype the command and press ENTER.

DEL

FUNCTION Deletes a file from disk.

COMMAND TYPE Internal

FORMAT

DEL [*drive:*][*path*]*filename* [*.ext*]

where the following is true:

drive: specifies the disk drive containing the file to delete. If you omit the drive qualifier, DEL uses the current default disk drive.

path specifies the name of the DOS directory that contains the file to be deleted. If you omit the path entry, DEL uses the current directory.

filename[.ext] is the name of the file to delete. DEL fully supports DOS wildcard characters.

NOTES DEL will not remove directories. Use RMDIR for that purpose.

Unless overridden with a drive or path specifier, DEL will delete files in the current directory only.

DOS allows you to delete several specific files at one time by placing each file name on the command line.

If you attempt to delete all of the files in a directory, DOS first prompts you with

```
A> DEL *.*
Are you sure (Y/N)?
```

to ensure that you want the command performed. If you want to delete the files, type **Y** and press ENTER; otherwise, type **N** and press ENTER.

WARNING: Once you have deleted a file, DOS cannot retrieve it.

EXAMPLES In the following example, DEL erases the contents of the CONFIG.BAK file from the current directory:

```
C> DEL CONFIG.BAK
```

The following command deletes a file from within a DOS subdirectory:

```
C> DEL \DOS\COMMANDS\AUTOEXEC.BAK
```

If you attempt to delete all of the files in the current directory, as shown here

```
C> DEL *.*
```

DEL responds with

```
Are you sure (Y/N)?
```

If you want to delete the files, type **Y** and press ENTER; otherwise, type **N** and press ENTER.

Lastly, to delete a file that resides on another disk, place a disk drive identifier in front of the file name to delete, as shown here:

```
C> DEL B:FILENAME.EXT
```

MESSAGES If DEL displays the message

```
Access Denied
```

you have tried either to delete a file marked as read-only by ATTRIB +R or DOS has located the specified file in a specified directory in the data-file search path (see APPEND).

If DEL displays the message

```
Are you sure (Y/N)?
```

you have issued a DEL command with the wildcard characters *.*. Since this command can have a devastating impact upon your system if invoked in error, DEL prompts you to verify your intentions.

If DEL displays the message

```
File not found
```

it could not find the specified file in the command line to delete.

If DEL displays the message

```
Invalid directory
```

a subdirectory in the path specifying the location of the file to delete is invalid.

If DEL displays the message

```
Invalid drive specification
```

the disk drive that precedes the path name of the file to delete is invalid. Make sure that you have included a colon after the disk-drive letter, and that the drive letter is valid (A, B, C, and so on).

DIR

FUNCTION Displays a directory listing of files.

COMMAND TYPE Internal

FORMAT

DIR [*file_specification*] [/P] [/W]

where the following is true:

file_specification is the complete DOS file specification for the file(s) for which DIR is to display the directory listing. It can contain a disk drive identification and path name. If you do not place a file specification in the command line, DIR displays a directory listing of all of the files in the current directory. DIR fully supports DOS wildcard characters.

/P directs DIR to pause after each screenful of information to display the prompt:

```
Press any key when ready...
```

/W directs DIR to display the files in short form (file name) only with five file names across the screen.

NOTES For each file, DIR will display its complete name, size in bytes, and the date and time of its creation or last modification.

DIR also displays the amount of free disk space in bytes.

To list all of the files on disk, including files in DOS directories, use the TREE command.

DIR does not display hidden system files.

DIR always displays the drive and directory in which the files are stored, as shown here:

```
Volume in drive A is DOSDISK
Directory of  A:\

SUBDIR          <DIR>        11-01-87      12:53p
            1 File(s)      1213440 bytes free
```

If you invoke DIR with only a file name, as shown here

```
C> DIR FILENAME
```

the extension defaults to *.

EXAMPLES In the following example, DIR displays a directory listing of each of the files in the current directory:

```
C> DIR
```

This command is functionally equivalent to

```
C> DIR *.*
```

If several files exist in the current directory, many may scroll off the screen during the directory listing. If this happens, invoke DIR with the /P qualifier, as shown here:

```
C> DIR /P
```

Each time DIR completes a screenful of files, it will pause and display the prompt

```
Press any key when ready...
```

When this occurs, press any key and DIR will continue.

The file specification of the DIR command can be quite specific, as shown here:

```
C> DIR \DOS\CONFIG.OLD
```

DIR displays the directory listing of the CONFIG.OLD file that resides in the DOS directory of the current disk.

If you have no interest in file sizes or their date and time stamps, invoke DIR with the /W option:

```
C> DIR /W
```

DIR displays a directory listing of file names only, five files across the screen, as shown here:

```
Volume in drive C is DOSDISK
Directory of C:\DOS
.          ..         ANSI     EXE  APPEND   COM    ASSIGN   COM
ATTRIB     EXE        BACKUP   COM  CHKDSK   COM    COMP     COM    DISKCOMP COM
DISKCOPY   COM        EDLIN    COM  FDISK    EXE    FIND     COM    FORMAT   COM
FORMATS    TBL        GRAFTABL COM  GWBASIC  EXE    HELP     EXE    JOIN     EXE
PRINT      COM        README        REPLACE  EXE    RESTORE  COM    SORT     EXE
           25 File(s)       7106560 bytes free
```

You can also use the DOS redirection operators with DIR. In this example, DOS displays a sorted directory listing:

```
C> DIR | SORT
```

This command prints the files in the current directory:

```
C> DIR > PRN:
```

To display the files contained on a disk other than the current default, place a disk drive specifier in the DIR command line, as shown here:

```
C> DIR A:
```

or

```
C> DIR A:FILENAME.EXT
```

MESSAGES If DIR displays the message

```
File not found
```

it could not find the file specified in the command line in the current directory or specified directory.

If DIR displays the message

```
Invalid directory
```

a subdirectory in the path specifying the location of the file to display a directory listing is invalid.

If DIR displays the message

```
Invalid drive specification
```

the disk drive that precedes the path name of the file to display a directory listing is invalid. Make sure that you have included a colon after the disk drive letter, and that the drive letter is valid (A, B, C, and so on).

If DIR displays the message

```
Strike a key when ready . . .
```

you have invoked DIR with the /P qualifier, which suspends the display of the directory listing with each screenful of information. Simply press a key to continue the directory listing, or press CTRL-BREAK to terminate the command.

If DIR displays the message

```
Invalid parameter
```

you have placed a qualifier other than /W (wide directory listing), or /P (screenful at a time) in your DIR command line.

DISKCOMP

FUNCTION Compares two floppy disks.

COMMAND TYPE External DISKCOMP.COM

FORMAT

[*drive:*][*path*]DISKCOMP [*primary_drive:* [*secondary_drive*]][/1] [/8]

where the following is true:

drive: specifies the disk drive that contains the DISKCOMP.COM file. If you omit the drive specifier DOS uses the current default disk drive.

path specifies the DOS directory name that contains the DISKCOMP.COM file. If you do not specify a path name, DOS uses the current default directory.

primary_drive: specifies one of the floppy-disk drives to be used in the disk comparison. If you do not specify a primary drive, DISKCOMP uses the current default disk drive.

secondary_drive specifies the second drive to be used in the disk comparison. If you do not specify a secondary drive, DISKCOMP uses the current default disk drive.

/1 directs DISKCOMP to perform a single-sided disk comparison.

/8 directs DISKCOMP to perform an eight-sector-per-track disk comparison.

NOTES If you have a single-floppy disk system, DISKCOMP performs a single-drive comparison, prompting you to enter the source and target disks at the correct time.

If the contents of the disks are identical, DISKCOMP displays the message

```
Compare OK
```

Otherwise, DISKCOMP displays the side and track (in hexadecimal form) of the differences.

DISKCOMP provides the following exit status values to support your batch processing.

0	Comparison OK
1	Disks were not the same
2	User has terminated using CTRL-C
3	Termination due to disk error
4	Insufficient memory or invalid disk drive

EXAMPLES In the following example, DISKCOMP will compare the contents of the disk in drive A to that in drive B:

```
C> DISKCOMP A: B:
```

If the disks are identical, DISKCOMP displays

```
Compare OK
```

Otherwise, DISKCOMP displays the locations of the differences, as shown here:

```
Compare error on
side n track n
```

If you need to compare single-sided disks, use this DISKCOMP command:

```
C> DISKCOMP A: B: /1
```

MESSAGES If DISKCOMP displays the message

```
Cannot DISKCOMP to or from
a Network drive
```

one of the disk drives that you have specified in the DISKCOMP command line references a network drive. DOS will not allow you to perform DISKCOMP operations to a network disk drive.

If DISKCOMP displays the message

```
Compare another diskette (Y/N)?
```

it has completed a disk comparison operation and is prompting you as to whether you want to perform a second disk comparison. If so, type **Y** and DISKCOMP will prompt you to enter the disks to compare. To terminate the DISKCOMP command, type **N** and press ENTER.

If DISKCOMP displays the message

```
Compare error on side n, track n
```

DISKCOMP has located differences between the two disks at the specified location. DISKCOMP compares two disks sector by sector, track by track, side by side. In this case, DISKCOMP has identified a difference in the specified track.

If DISKCOMP displays the message

```
Compare OK
```

the two disks are identical.

If DISKCOMP displays the message

```
Comparing n tracks
n sectors per track, n side(s)
```

it is telling you the format of the disks that it is comparing. DISKCOMP compares two disks sector by sector, track by track, side by side. This message tells you the number of tracks, sides, and sectors that DISKCOMP will examine on each disk.

If DISKCOMP displays the message

```
Drive n not ready
Make sure a diskette is inserted into
the drive and the door is closed
```

the disk drive latch on the specified drive is probably open. Make sure that you have inserted a disk into the drive and that the disk drive door is closed. Repeat the DISKCOMP command.

If DISKCOMP displays the message

```
First diskette bad or incompatible
```

the disk in the first drive specified in the DISKCOMP command line is either damaged or incompatible with the disk drive in which it resides. For example, DISKCOMP will not allow you to examine a 1.2 MG disk in a 360K floppy-disk drive.

If DISKCOMP displays the message

```
Second diskette bad or incompatible
```

the disk in the second drive specified in the DISKCOMP command line is either damaged or incompatible with the disk drive in which it resides.

If DISKCOMP displays the message

```
Incorrect DOS version
```

the executable DISKCOMP.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct DISKCOMP.COM for the executing version of DOS.

If DISKCOMP displays the message

```
Insert diskette with directory that contains
COMMAND.COM in drive n and strike any key when ready
```

you need to insert a disk containing COMMAND.COM in the root directory before DOS can continue. Remember that COMMAND.COM contains your internal DOS commands and the command processor responsible for displaying the DOS prompt. If this message occurs, insert the disk that you use to boot DOS into the specified drive and strike a key.

If DISKCOMP displays the message

```
Insert FIRST diskette in drive n:
Press any key when ready ...
```

it is prompting you to insert one of the two disks to compare into the specified drive. If you are performing a single-disk drive comparison, you will need to exchange floppy disks within the specified drive. One disk will have to be the first, while another will be the second. In this case, DISKCOMP is prompting you to enter the first disk into the drive.

If DISKCOMP displays the message

```
Insert SECOND diskette in drive n:
Press any key when ready ...
```

it is prompting you to insert the second of the two disks to compare into the specified drive.

If DISKCOMP displays the message

```
Insufficient memory
```

your system has insufficient memory to execute the DISKCOMP command. If you have installed memory-resident software or a RAM drive, consider restarting DOS without them so that you can repeat the DISKCOMP command.

If DISKCOMP displays the message

```
Invalid drive specification
Specified disk does not exist
or is non-removable
```

the disk drive that you specified in your DISKCOMP command line is either invalid or a fixed disk. DISKCOMP does not allow you to compare fixed disks. If the specified drive is not a fixed disk, make sure that you have placed a colon after the disk drive letter, and that you are using a valid floppy-disk drive letter (A, B, and so on).

If DISKCOMP displays the message

```
Invalid parameter
Do not specify filename(s)
Command format: DISKCOMP d: d: [/1] [/8]
```

your DISKCOMP command line is invalid. Examine the DISKCOMP command format shown, and reissue the command.

DISKCOPY

FUNCTION Copies a source floppy disk to a target disk.

COMMAND TYPE External DISKCOPY.COM

FORMAT

[*drive:*][*path*]DISKCOPY [*source_drive:* [*target_drive*]][/1]

where the following is true:

drive: specifies the disk drive containing the DISKCOPY.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path specifies the DOS directory that contains the DISKCOPY.COM file. If you do not specify a path name, DOS uses the current default directory.

source_drive: specifies the disk drive containing the floppy disk to copy. If you do not specify a source drive, DISKCOPY uses the current default disk drive.

target_drive specifies the disk drive containing the disk to be copied to. If you do not specify a target drive, DISKCOPY uses the current default disk drive.

/1 directs DISKCOPY to copy only the first side of the source disk to the target disk.

NOTES DISKCOPY copies the contents of one floppy disk to another. If you have a single-floppy disk system, DISKCOPY performs a single-drive copy, prompting you to enter the source and target disks at the correct time.

DISKCOPY formats an unformatted disk during the copy.

DISKCOPY does not correct disk fragmentation. Use the DOS XCOPY command for that purpose.

DISKCOPY destroys the previous contents of the target disk.

If the source and target disks are not the same type (for example, a 360K and 1.2MB disk), DISKCOPY displays a message and terminates.

Do not use DISKCOPY in conjunction with JOIN or SUBST.

If the target disk has not been formatted, DISKCOPY formats the disk during the copy operation, displaying the message

```
Formatting while copying
```

DISKCOPY provides the following exit status code values to support your batch processing:

0	Successful copy
1	Unsuccessful due to nonfatal disk error
2	User has terminated by using CTRL-C
3	Unsuccessful due to fatal disk error
4	Insufficient memory or invalid disk drives

EXAMPLES The following command assumes that you have two compatible floppy-disk drives on your system:

```
C> DISKCOPY A: B:
```

When DISKCOPY begins copying a disk, it reads several tracks of data from the source and then writes them to the target disk. In a single-floppy disk drive system, DISKCOPY repeats this process and prompts you for the source and target disks, as shown here:

```
[A:\] DISKCOPY
Insert SOURCE diskette in drive A:

Press any key when ready ...

Copying 40 tracks
9 Sectors/Track, 2 Side(s)

Insert TARGET diskette in drive A:
```

```
Press any key when ready ...

Insert SOURCE diskette in drive A:

Press any key when ready ...

Insert TARGET diskette in drive A:

Press any key when ready ...

Copy another diskette (Y/N)?
```

MESSAGES If DISKCOPY displays the message

```
Cannot DISKCOPY to or from
a Network drive
```

one of the disk drives specified in the DISKCOPY command line references a network disk drive. DOS will not allow you to use a network device in an operation involving DISKCOPY.

If DISKCOPY displays the message

```
Copying n tracks
n Sectors per Track, n Side(s)
```

it is telling you the format of the disk that it is duplicating. DISKCOPY copies one disk to another sector by sector, track by track, and side by side.

If DISKCOPY displays the message

```
Copy process ended
```

the DISKCOPY operation has failed. Normally, DISKCOPY will precede this error message with a message describing the reason for the failure.

If DISKCOPY displays the message

```
Disk error while reading drive n:
Abort, Retry, Fail?
```

the disk drive latch for the specified drive is likely to be open. If so, close the drive latch and select the R (Retry) option.

If DISKCOPY displays the message

```
Drive n not ready
Make sure a diskette is inserted into
that drive and the door is closed
```

the disk drive latch for the specified drive is very likely still open. If so, close the drive latch and select the R (Retry) option.

If DISKCOPY displays the message

```
Drive types or diskette types
not compatible
```

either the disks or drives specified in the DISKCOPY command are incompatible. DISKCOPY will not, for example, allow you to copy the contents of a 1.2MB disk to a 360K disk.

If DISKCOPY displays the message

```
Formatting while copying
```

the target disk in the DISKCOPY operation was unformatted. DISKCOPY must format the disk before it can be used by DOS. DISKCOPY is informing you that it is formatting the disk before performing the copy operation.

If DISKCOPY displays the message

```
Incorrect DOS version
```

the executable DISKCOPY.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct DISKCOPY.COM for the executing version of DOS.

If DISKCOPY displays the message

```
Insert diskette with directory that contains
COMMAND.COM in drive n and strike any key when ready
```

you need to insert a disk containing COMMAND.COM in the root directory before DOS can continue. Remember that COMMAND.COM contains your internal DOS commands and the command processor responsible for displaying the DOS prompt. If this message occurs, insert the disk that you use to boot DOS into the specified drive and strike a key.

If DISKCOPY displays the message

```
Insert SOURCE diskette in drive n:
Press any key when ready...
```

DISKCOPY is prompting you to insert the disk that you want to copy (the source disk) into the specified drive. Once you do so, simply press any key to continue the DISKCOPY operation.

If DISKCOPY displays the message

```
Insert TARGET diskette in drive n:
Press any key when ready...
```

DISKCOPY is prompting you to insert the disk that you want it to copy to (the target disk) into the specified drive. Once you do so, press any key to continue the DISKCOPY operation.

If DISKCOPY displays the message

```
Insufficient memory
```

your system does not have enough available RAM to execute the DISKCOPY command. If you have installed memory-resident software or a RAM drive, consider restarting DOS without them so that you can execute the DISKCOPY command.

If DISKCOPY displays the message

```
Invalid parameter
Do not specify filename(s)
Command Format: DISKCOPY D: D: [/1]
```

your DISKCOPY command line contains invalid entries. Examine the format shown and reissue the correct DISKCOPY command.

If DISKCOPY displays the message

```
SOURCE diskette bad or incompatible
```

the source disk (the disk to copy) is damaged or incompatible with the disk drive it resides in. Make sure that you are not trying to use a 1.2MB disk in a 360K disk drive and reissue the DISKCOPY command. If the command fails again, your source disk may be severely damaged.

If DISKCOPY displays the message

```
TARGET diskette bad or incompatible
```

the target disk (the disk to copy to) is damaged or incompatible with the disk drive it resides in. Make sure that you are not trying to use a 1.2MB disk in a 360K disk drive and reissue the DISKCOPY command. If the command fails again, your target disk may be severely damaged.

If DISKCOPY displays the message

```
Target diskette is write protected
```

the target disk in the DISKCOPY command contains a write-protect tab that prevents DISKCOPY from writing to it. In most cases, this error occurs when you have forgotton to place the target disk in the drive, and DISKCOPY is attempting to overwrite a different disk. Write protecting your disks can protect them from many errant DOS commands.

If DISKCOPY displays the message

```
Target diskette may be unusable
```

it was unable to perform an identical disk copy of the source disk. This error normally occurs when the target disk contains damaged locations that prevent their use. To ensure that you have a complete disk copy, repeat the DISKCOPY command with a different target disk.

If DISKCOPY displays the message

```
Unrecoverable read error on drive n:
side n, track n
```

it was unable to read the specified track of the source disk. Since DISKCOPY copies one disk to another sector by sector, track by track, your disk copy will be incomplete due to the missing information. In most cases, this error is associated with write-protected software or a damaged source disk.

If DISKCOPY displays the message

```
Unrecoverable write error on drive n:
side n, track n
```

it was unable to write to the specified track of the target disk. Since DISKCOPY copies one disk to another sector by sector, track by track, your disk copy will be incomplete due to the missing information. In most cases, this error is associated with a damaged target disk.

ECHO

FUNCTION Displays or suppresses batch command messages.

COMMAND TYPE Internal

FORMAT

ECHO [ON | OFF | *message*]

where the following is true:

ON enables the display of batch commands as they execute.

OFF disables the display of batch commands as they execute.

message is the text message ECHO is to display to the user.

NOTES By default, each time you execute DOS batch files, DOS displays the names of each command as it executes. For example, the file

```
A
B
C
```

displays the following:

```
C>A

C>B

C>C
```

ECHO allows you to suppress the display of command names within a DOS batch file as the command executes. For example, the file

```
ECHO OFF
A
B
C
```

will display

```
C> ECHO OFF
```

ECHO also provides a convenient method of displaying messages to the user.

If you invoke ECHO without a command-line parameter, ECHO displays its current state, ON or OFF.

If you are using DOS version 3.3, you can precede each command with the @ character to suppress display of the command name.

EXAMPLES In the following command, DOS continues to display command names within a DOS batch file as it executes:

```
C> ECHO ON
```

This command disables command-name display:

```
C> ECHO OFF
```

ECHO can be used with batch parameters. For example, this command displays the name of the batch file that is currently executing:

```
ECHO %0
```

This batch procedure displays each of its command-line parameters:

```
ECHO OFF
:LOOP
SHIFT
IF '%0'== '' GOTO DONE
ECHO %0
GOTO LOOP
:DONE
```

This ECHO command displays a simple message to the user:

```
ECHO This is a message via ECHO
```

The following batch file fully exploits ECHO, displaying copyright information on the screen:

```
ECHO OFF
ECHO ****************************************
ECHO * Kevin Shafer Publishing Inc. 1988  *
ECHO *                                    *
ECHO * Raiders Football Summary          *
ECHO *                                    *
ECHO ****************************************
```

When invoked, the procedure will display

```
ECHO OFF

*********************************************
* Kevin Shafer Publishing Inc. 1988         *
*                                           *
* Raiders Football Summary                  *
*                                           *
*********************************************
```

Under DOS version 3.3, you can suppress the display of ECHO OFF by using the line

```
@ECHO OFF
```

in the batch procedure shown previously, with and without the ECHO OFF command, to better understand the impact of this command.

ERASE

FUNCTION Erases a file from disk.

COMMAND TYPE Internal

FORMAT

ERASE [*drive:*][*path*]*filename* [*.ext*]

where the following is true:

drive: specifies the disk drive containing the file to be deleted. If you omit the drive qualifier, ERASE uses the current default disk drive.

path specifies the name of the DOS directory that contains the file to be deleted. If you omit the path entry, ERASE uses the current directory.

filename[.ext] is the name of the file to be deleted. ERASE fully supports DOS wildcard characters.

NOTES ERASE does not remove directories. Use RMDIR for that purpose.

ERASE and DEL are functionally equivalent. Most people use DEL.

Unless overridden with a drive or path specifier, ERASE deletes only files in the current directory.

If you attempt to erase all of the files in a directory, DOS first prompts you with

```
Are you sure (Y/N)?
```

to verify that you want the command performed. If you want to delete the files, type **Y** and press ENTER; otherwise, type **N** and press ENTER.

WARNING: Once you have erased a file, DOS cannot retrieve it.

EXAMPLES In the following example, ERASE erases the contents of the TEST.DAT file from drive B:

```
C> ERASE B:TEST.DAT
```

The following command erases a file from within a DOS subdirectory:

```
C> ERASE \DOS\COMMANDS\AUTOEXEC.BAK
```

If you attempt to delete all of the files in the current directory, as shown here

```
C> ERASE *.*
```

ERASE will respond with

```
Are you sure (Y/N)?
```

If you want to delete the files, type **Y** and press ENTER; otherwise type **N** and press ENTER.

To erase a file from a disk drive other than the current default, precede the file name with the disk drive letter, as shown here:

```
C> ERASE A:FILENAME.EXT
```

EXE2BIN

FUNCTION Converts an EXE file format to a COM file.

COMMAND TYPE External EXE2BIN.EXE

FORMAT

[*drive:*][*path*]EXE2BIN *source_file target_file*

where the following is true:

drive: specifies the disk drive containing the EXE2BIN.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path specifies the DOS directory that contains the EXE2BIN.COM file. If you do not specify a path name, DOS uses the current default directory.

source_file is the name of the EXE file that DOS is to convert to a COM file.

target_file is the name of the converted COM file.

NOTES If you do not specify a file extension on the source file, EXE2BIN assumes EXE. If you do not specify a file extension on the target file, EXE2BIN assumes BIN.

If a drive or path is not specified for either file, EXE2BIN uses the current defaults.

COM files contain memory images of the program. Therefore, they result in a more compact file, which DOS can load quickly into memory. However, there are several requirements that the EXE file must meet before it can be converted. For more information on this, refer to the *DOS Technical Reference Manual.* This is a programmer's tool. Most users will never execute this command.

EXAMPLE The following command converts the FILENAME.EXE file to FILENAME.COM:

```
C> EXE2BIN FILENAME.EXE FILENAME.COM
```

MESSAGES If EXE2BIN displays the message

```
Incorrect DOS version
```

the executable EXE2BIN.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct EXE2BIN.EXE for the executing version of DOS.

If EXE2BIN displays the message

```
Access denied
```

the file that you are attempting to create already exists on disk and is write protected.

If EXE2BIN displays the message

```
File not found
```

DOS could not find the specified file in the command line that EXE2BIN is to convert to a binary file.

If EXE2BIN displays the message

```
Insufficient memory
```

your system has too little RAM to execute the program. If you have installed memory-resident programs or a RAM drive, consider removing them and repeating the EXE2BIN command.

If EXE2BIN displays the message

```
File creation error
```

either a subdirectory already exists on disk with the same name as the target file of the EXE2BIN command or the root directory of the target disk is full.

If EXE2BIN displays the message

```
Amount read less than size in header
```

the size of the executable file read does not correspond to the size specified in the executable file header.

EXIT

FUNCTION Terminates a secondary command processor.

COMMAND TYPE Internal

FORMAT

```
EXIT
```

NOTES EXIT allows you to terminate a secondary command processor.

Many software packages (such as word processors and spreadsheets) allow you to access the DOS prompt by creating a secondary process from your application. Once you are done issuing your DOS commands, invoke EXIT to return to your application.

EXAMPLE The following command terminates a secondary command processor:

```
C> EXIT
```

MESSAGES If EXIT displays the message

```
Bad command or filename
```

you are using a DOS version earlier than 2.0. DOS introduced the EXIT command with version 2.0.

FASTOPEN

FUNCTION Increases the number of directory entries that DOS keeps in memory, thus increasing the speed at which DOS locates files on disk.

COMMAND TYPE External FASTOPEN.EXE

FORMAT

[*drive:*][*path*]FASTOPEN *d:* [=*entries*] [...]

where the following is true:

drive: specifies the disk drive containing the FASTOPEN.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path specifies the DOS directory that contains the FASTOPEN.COM file. If you do not specify a path name, DOS uses the current default directory.

d: specifies the disk drive identification for which DOS is setting aside storage to contain directory entries.

entries specifies the number of directory entries that DOS is reserving space for. This value must be in the range from 10 to 999. The default is 10.

... indicates that FASTOPEN can reserve space for several disks in one command.

NOTES FASTOPEN is a DOS version 3.3 command.

When you specify several disks, remember that the sum of the entries cannot exceed 999.

Each directory entry stored requires 35 bytes.

Each time DOS accesses a directory, it places that directory name into a list of the directory names. Each time you perform a directory manipulation command, DOS first checks its list of directories in memory to see if it can locate the directory on disk instead of reading the disk in search of the directory.

If you use the value of 999 for directory entries, DOS spends a considerable amount of overhead searching its directory list. Most users, therefore, find the default value of 10 acceptable.

EXAMPLE In the following command, DOS remembers 50 directory entries for drive C:

```
C> FASTOPEN C:=50
```

MESSAGES If FASTOPEN displays the message

```
Cannot use FASTOPEN for drive n:
```

either the specified disk drive is a floppy-disk drive or used in conjunction with ASSIGN or SUBST. FASTOPEN supports only fixed disks.

If FASTOPEN displays the message

```
FASTOPEN installed - memory resident software
```

it is telling you that it has successfully loaded itself in memory and is ready to assist DOS in opening your files.

If FASTOPEN displays the message

```
FASTOPEN already installed
```

it is telling you that it has already been installed in memory. Since FASTOPEN installs memory-resident software, it can be invoked only once per user session. To change FASTOPEN parameters, you must restart DOS.

If FASTOPEN displays the message

```
Incorrect number of parameters
```

you have likely not included a disk drive specifier in your FASTOPEN command line. FASTOPEN requires you to specify each disk drive letter that you want it to support.

If FASTOPEN displays the message

```
Insufficient memory
```

your system does not have enough RAM for FASTOPEN to successfully execute. If you have installed other memory-resident software or a RAM disk, you may want to reinvoke DOS without them, or specify fewer directory entries in the FASTOPEN command line. Each entry consumes 35 bytes.

If FASTOPEN displays the message

```
Invalid drive specified
```

one of the disk drives specified in your command line is invalid. Make sure that you have placed a colon after the disk drive name and that you are using existing disk drives (A, B, C, and so on). Remember, FASTOPEN supports only fixed-disk drives.

If FASTOPEN displays the message

```
Invalid parameter
```

make sure that you have placed an equal sign between each disk drive letter and the desired number of directory entries.

If FASTOPEN displays the message

```
Same drive specified more than once
```

you have placed the same disk drive letter twice in your FASTOPEN command line. Correct the error and reissue the command.

If FASTOPEN displays the message

```
Too many drive entries
```

invoke the FASTOPEN command with fewer disk drives. Specify only your most commonly used disk drives.

If FASTOPEN displays the message

```
Too many name entries
```

the total number of directory entries in the FASTOPEN command line exceeds 999. Reduce this number and reissue the command. If you specify too many entries, FASTOPEN may reduce your system's performance since it must spend considerable time checking each entry. Also, each FASTOPEN entry consumes 35 bytes.

If FASTOPEN displays the message

```
Incorrect DOS version
```

the executable FASTOPEN.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct FASTOPEN.EXE for the executing version of DOS. Remember, FASTOPEN does not exist for DOS versions earlier than 3.3.

FC

FUNCTION Compares two files and displays the differences between them.

COMMAND TYPE External FC.EXE

FORMAT

[*drive:*][*path*]FC [/A][/B][/C][/L b*n*][/N][/T][W][/*nnnn*]
file1 *file2*

where the following is true:

drive: specifies the disk drive of the disk containing the FC.EXE file. If you do not specify a drive identification, DOS uses the default disk drive.

path specifies the directory containing the FC.EXE file. If you do not specify a subdirectory path, DOS uses the current default directory.

/A directs FC not to display all of the lines that differ between two ASCII files, but rather the first and last line in a group of lines that differ.

/B directs FC to perform a binary comparison of the files.

/C directs FC to ignore the difference between upper- and lowercase letters.

/L directs FC to perform an ASCII file comparison.

/Lb *n* directs FC to use an internal buffer of *n* lines.

/N directs FC to display the line numbers of lines that differ in two ASCII files.

/T directs FC not to expand tabs into spaces.

/W directs FC to compress white space (spaces or tabs).

/*nnnn* specifies the number of lines that must again match following a difference for FC to assume the two files are synchronized. The default is 2.

file1 and *file2* are the files to be compared.

NOTES FC displays differences between two files by displaying the first file name followed by the lines that differ between the files. Next, FC displays the second file name followed by the first line that matches in both files and the differences.

If FC is comparing binary files, it displays differences in the form

aaaaaaaa bb cc

where *aaaaaaaa* is the offset address of the values that differ, while *bb* and *cc* are the values that differ.

If FC encounters too many errors, it terminates and displays the message

```
Resynch failed. Files are too different.
```

FC is often a more convenient command to use when comparing two files than is the COMP command.

EXAMPLES The following command compares two binary files, using the /B qualifier:

```
C> FC TEST.EXE OLDTEST.EXE
```

The output might be

```
00000002: 9A 66
00000004: 06 20
00000006: 01 03
0000000A: 00 CE
0000000E: 20 14
0000000F: 00 04
00000011: 02 08
00000012: 32 97
```

Assuming that the A.DAT file contains

```
1
2
3
4
5
A
A
A
6
7
8
9
10
```

and B.DAT contains

```
1
2
3
4
5
B
B
B
6
7
8
9
10
```

the command

```
C> FC A.DAT B.DAT
```

displays

```
***** A.DAT
5
A
A
A
6
***** B.DAT
5
B
B
B
6
*****
```

If you include the /N qualifier, as shown here

```
C> FC /N A.DAT B.DAT
```

FC displays the following:

```
***** A.DAT
5: 5
6: A
7: A
8: A
9: 6
***** B.DAT
5: 5
6: B
7: B
8: B
9: 6
*****
```

If the DAYS.DAT file contains

```
SUNDAY
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
SATURDAY
```

and WEEKDAYS.DAT contains

```
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
```

the command

```
C> FC DAYS.DAT WEEKDAYS.DAT
```

displays

```
***** DAYS.DAT
SUNDAY
MONDAY
TUESDAY
***** WEEKDAYS.DAT
MONDAY
TUESDAY
*****

***** DAYS.DAT
SATURDAY
***** WEEKDAYS.DAT
*****
```

If you want FC to ignore differences between upper- and lowercase letters, add the /C qualifier, as shown here:

```
C FC /C FILE.1 FILE.2
```

MESSAGES The following error messages are common to this command. Use this list of messages to resolve errors you experience when you issue commands from the DOS prompt.

If FC displays the message

```
FILENAME.EXT larger than FILENAME.EXT
```

the contents of each file compared identically; however, the first file name specified is simply longer than the second file.

If FC displays the message

```
cannot open FILENAME.EXT - no such file or directory
```

FC could not locate the file that you specified in the command. Check your spelling of the file name and current directory and reissue the FC command.

If FC displays the message

```
Incompatible switches
```

you have placed the /B (binary file switch) in the same command line as a ASCII file qualifier.

If FC displays the message

```
Incorrect DOS version
```

the executable FC.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct FC.EXE for the executing version of DOS.

If FC displays the message

```
no differences encountered
```

both of the files compared identically.

If FC displays the message

```
Out of memory
```

your system has insufficient RAM for FC to create the tables that it requires to store the information from each file during the comparision. If you have installed memory-resident software or a RAM drive, you might consider removing them so that you can repeat the FC command.

If FC displays the message

```
Resynch failed Files are too different
```

the files are so different that FC could not locate a position in both files that were similar enough to resume the file comparison.

If FC displays the message

```
usage: fc [/a] [/b] [/c] [/l] [/lbNN] [/w] [/t] [/n] [/NNNN] file1    file2
```

you invoked FC without any command line parameters and FC is displaying its optional command-line parameters.

FDISK

FUNCTION Defines disk partitions on a fixed disk.

COMMAND TYPE External FDISK.COM

FORMAT

```
FDISK
```

NOTES DOS allows you to divide your fixed disk into logical collections of cylinders, known as partitions. This allows you to place several different operating systems on one fixed disk.

The DOS FDISK command allows you to add, change, display, and delete disk partitions.

The first sector on every fixed disk contains a master boot record. This record contains partition information. This defines which partition the computer uses to boot from. FDISK is your interface to the master boot record.

EXAMPLES FDISK is a menu-driven program. If you invoke FDISK with

```
C> FDISK
```

it responds by displaying the first menu, as shown here:

```
Fixed Disk Setup Program Version 3.30
(C)Copyright Microsoft Corp. 1987

FDISK Options

Current Fixed Disk Drive: 1
```

If you want to create your first fixed disk partition (drive C), select option 1. Once you define this partition, you can exit back to DOS and then use FORMAT to prepare your fixed disk.

If you have a disk larger than 30MB and want to create multiple partitions, select option 2 after you have defined your primary DOS partition.

To create two DOS partitions, a primary and an extended partition, select option 4. FDISK displays

```
Display Partition Information

Current Fixed Disk Drive: 1

Partition Status  Type     Start     End Size
C: 1          A    PRI DOS  0       523 524
2             EXT DOS     524       964 441


Total disk space is  965 cylinders.

The Extended DOS partition contains
logical DOS drives. Do you want to
display logical drive information? [Y]


Press ESC to return to FDISK Options
```

MESSAGES If FDISK displays the message

```
DOS partition created
```

FDISK has successfully created your DOS partition as specified. Once you exit FDISK, you must format the disk before it can be used by DOS.

If FDISK displays the message

```
Do you wish to use the entire fixed
disk for DOS (Y/N).....?[Y]
```

FDISK is prompting you as to whether or not you want to use the entire disk as a DOS partition. In most cases, most users will want to use their entire disk as a DOS partition.

If FDISK displays the mesage

```
Do you wish to use the maximum size nnn
for a DOS partition and make the DOS
partition active (Y/N)...?
```

FDISK is prompting you as to whether or not you want to use the entire disk as a DOS partition. In most cases, most users will want to use their entire disk as a DOS partition.

If FDISK displays the message

```
Enter the number of the partition you
want to make active.....[n]
```

it is prompting you to enter the number associated with the disk partition that you want to make active. Although FDISK allows you to logically divide your disk into several partitions, only one partition can be active (or bootable). In most cases, you will keep your DOS partition as the active partition.

If FDISK displays the message

```
Fixed disk already has a DOS partition
```

you have selected the Create DOS partition when a DOS partition already exists. You can only place one active DOS partition on a fixed disk.

If FDISK displays the message

```
Incorrect DOS version
```

the executable FDISK.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct FDISK.COM for the executing version of DOS.

If FDISK displays the message

```
Maximum space available for partition n
is nnn cylinders
```

you have instructed FDISK not to use the entire disk for the partition that you are creating. As such, FDISK is telling you the maximum size of the partition that you can create.

If FDISK displays the message

```
No DOS partition to delete
```

you have selected the FDISK menu option Delete DOS partition and the disk does not contain a DOS partition that DOS can delete.

If FDISK displays the message

```
No fixed disk present
```

either your system has no fixed disk installed or your system is experiencing a hardware error. Turn off the power of your computer and restart DOS to see if the error corrects itself. If the error does not disappear, call your manufacturer.

If FDISK displays the message

```
No partitions defined
```

you have selected the Display Partition Information option and you have not yet created partitions on the fixed disk.

If FDISK displays the message

```
No partition to make active
```

you have selected the Change Active Partition option and you have not yet created partitions on the fixed disk that FDISK can make active.

If FDISK displays the message

```
No space to create a DOS partition
```

you have selected the Create DOS partition option and your disk does not have sufficient available free cylinders to create a DOS partition. If this error occurs, you may have to modify the size of the non-DOS partitions that you have created on the disk.

If FDISK displays the message

```
Partition n is already active
```

you have selected the Change Active Partition option and the partition that you specified was already active.

If FDISK displays the message

```
Partition n made active
```

you have selected the Change Active Partition option and specified a new boot partition. This message is purely informational.

If FDISK displays the message

```
System will now restart
Insert DOS diskette in drive A
Press any key when ready ...
```

you have successfully created a DOS partition and FDISK is ready to restart your system so that you can format your newly created fixed disk. Place a bootable disk in drive A and press any key to continue.

If FDISK displays the message

```
Total disk space is nnn cylinders
Maximum available space is nnn
cylinders at nn
```

FDISK is telling you the current disk utilization and maximum space that you can use to create an additional disk partition.

If FDISK displays the message

```
Warning! Data in the DOS partition
will be lost. Do you wish to
continue.....? [N]
```

it is prompting you to specify whether or not you really want to delete the current DOS partition, warning you that if you do so, all of the data that the partition contains will be lost. If you simply press ENTER, FDISK will select the default option of no ([N]).

If FDISK displays the message

```
Warning! Data in the Extended DOS
partition will be lost. Do you wish to
continue.....? [N]
```

FDISK is prompting you to specify whether or not you really want to delete an extended DOS partition, warning you that if you do so, all of the data that the partition contains will be lost. To delete the extended DOS partition, type **Y** and press ENTER. If you don't want to delete the extended DOS partition, simply press ENTER and DOS will select the default of no.

If FDISK displays the message

```
Cannot FDISK with network loaded
```

FDISK is telling you that you cannot execute the FDISK command when your system is part of a local-area network. If you want to execute the FDISK command, you must terminate the network program.

If FDISK displays the message

```
Requested partition size exceeds the
maximum available space
```

the number of cylinders that you have directed FDISK to use for your disk partition exceeds the number of available cylinders. Repeat the option specifying a smaller number of cylinders.

If FDISK displays the message

```
Partition selected is not bootable
active partition not changed
```

you have selected the Change Active Partition option and specified a partition that cannot be booted. As such, FDISK has left the active partition unchanged.

If FDISK displays the message

```
Cannot create an extended DOS partition
without a primary DOS partition on disk 1
```

FDISK is informing you that you cannot create an extended DOS partition on a disk drive until you create a primary (bootable) DOS partition on the same

disk. Many users who have disks that are larger than 30MB will create a 30MB primary DOS partition as drive C and then extended DOS partitions in the remaining disk space.

FIND

FUNCTION Searches a file or piped input for a string.

COMMAND TYPE External FIND.EXE

FORMAT

[*drive:*][*path*]FIND [/C][/N][/V] "*string*" [*file_spec* ...]

where the following is true:

drive: specifies the disk drive containing the FIND.COM file. If you do not specify a disk drive id, DOS uses the current default disk drive.

path specifies the name of the DOS directory that contains the FIND.COM file. If you do not specify a directory path, DOS uses the current default directory.

string specifies the string for which FIND is to search. It must be in quotes.

file_spec is the file name that is to be searched for the string. It can be a series of file names separated by spaces. FIND does not support DOS wildcard characters.

... indicates that several file names can reside in the command line.

/C directs FIND to display a count of occurrences of the string.

/N directs FIND to precede each line containing the string with its line number.

/V directs FIND to display each line that does not contain the string.

NOTES FIND allows you to quickly locate a sequence of characters within a file or redirected output. FIND can also be used as a filter of piped input.

If /C and /N are used together, FIND ignores /N.

The string FIND is searching for must be in quotes. If the string has nested quotes within, you must use two quotes at each nested quote, such as

```
C> FIND """Watch out!"" Dave yelled" FILENAME.EXT
```

EXAMPLES In the following example FIND is a filter used to list each directory in the current directory:

```
C> DIR A: | FIND "<DIR>"
```

If the current directory contains

```
Volume in drive A is DOSDISK
Directory of  A:\

SUBDIR        <DIR>                11-01-87    12:53p
A             <DIR>                11-01-87    1:44p
B             <DIR>                11-01-87    1:44p
C             <DIR>                11-01-87    1:44p
WSSTRIP       EXE          2816    10-03-84    10:33p
TAB           EXE          12326   5-05-85     6:25p
SUM           EXE          2834    10-24-87    2:20p
PIPE          EXE          2814    10-26-87    6:47p
PIPE2         EXE          7528    10-23-87    12:37p
          9 File(s)     1182208 bytes free
```

the previous command displays

```
SUBDIR        <DIR>        11-01-87    12:53p
A             <DIR>        11-01-87    1:44p
B             <DIR>        11-01-87    1:44p
C             <DIR>        11-01-87    1:44p
```

To list all files that are not directories, use the FIND /V option, as shown here:

```
C> DIR A: | FIND /V "<DIR>"
```

The following command displays each occurrence of the string "begin" in the TEST.PAS file:

```
C> FIND "begin" TEST.PAS
```

This command displays each occurrence of the string "begin," and each line is preceded by its line number:

```
C> FIND /N "begin" TEST.PAS
```

The following command displays a count of the number of occurrences of "begin" in the file:

```
C> FIND /C "begin" TEST.PAS
```

MESSAGES If FIND displays the message

```
FIND: Access denied - FILENAME.EXT
```

it could not access the specified file because it is locked by another file in the network.

If FIND displays the message

```
FIND: File not found - FILENAME.EXT
```

it could not locate one of the files specified in your FIND command line. Check your spelling and the current directory and reissue the FIND command.

If FIND displays the message

```
FIND: Invalid number of parameters
```

your FIND command line does not specify at least a search string for FIND to look. You must specify at least FIND "string".

If FIND displays the message

```
FIND: Invalid parameter - n
```

the specified qualifier is invalid. Review the format of the FIND command and reissue the command.

If FIND displays the message

```
FIND: Read error in FILENAME.EXT
```

the specified file is very likely corrupted. Repeat the command to see if the error goes away. If you do not have a backup copy of the file, you may have to use the RECOVER command to save a portion of the file.

If FIND displays the message

```
FIND: Syntax error
```

your FIND command format is invalid. Make sure that you have specified the search string and qualifiers in the right order.

If FIND displays the message

```
Incorrect DOS version
```

the executable FIND.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct FIND.EXE for the executing version of DOS.

FOR

FUNCTION Provides repetitive execution (iterative processing) of DOS commands.

COMMAND TYPE Internal

FORMAT

FOR %%*variable* IN (*set*) DO *DOS_command*

where the following is true:

%%variable is the FOR loop control variable that DOS manipulates with each iteration. The variable name is restricted to a character, but 0-9 cannot be used because they are reserved for DOS batch parameters.

set is a list of valid DOS file names. It can be a list of DOS file names separated by commas (A, B, C), or it can contain a wildcard character (*.*), or both (A, B, *.DAT).

DOS_command is the command to execute with each iteration.

NOTES FOR is most commonly used within DOS batch files. However, FOR can be used from the DOS prompt. The %% symbols before the variable name are used in batch files, while the % symbol is used from the DOS prompt.

EXAMPLES The set processing that FOR performs is straightforward. Consider this example:

```
FOR %%V IN (AUTOEXEC.BAT, TEST.DAT, CONFIG.SYS) DO TYPE %%V
```

FOR assigns the file name AUTOEXEC.BAT to the variable during the first iteration and displays its contents. On the second iteration, FOR assigns the variable the file name TEST.DAT and displays its contents. On the third iteration, FOR assigns the file name CONFIG.SYS to the variable and again displays the file's contents. When FOR prepares for the fourth iteration it fails to find any more file names, so its processing is complete.

The following command has been issued from the DOS prompt. Notice that the variable name is preceded by only one % symbol:

```
C> FOR %I IN (*.FOR) DO TYPE %I
```

In this command, a DOS batch file is using FOR to compile all of the C files on the current directory:

```
FOR %%F IN (*.C) DO CC %%F
```

MESSAGES The following error messages are common to this command. Use this list of messages to resolve errors you experience when you issue commands from the DOS prompt.

If FOR displays the message

```
FOR cannot be nested
```

you have placed a second FOR command as the DOS command that FOR will execute with each iteration, which is called nesting. FOR does not allow you to specify the FOR command as its command to execute.

If FOR displays the message

```
Syntax error
```

your command format is incorrect. Make sure that your command line contains the word DO.

FORMAT

FUNCTION Formats a disk for use by DOS.

COMMAND TYPE External FORMAT.COM

FORMAT

[*drive:*][*path*]FORMAT [*d:*][/S][/V][/4][/t:*tracks*]
[/n:*sectors*][/1][/8][/B]

where the following is true:

drive: is the disk drive containing the FORMAT.COM file. If you omit the disk drive specifier, DOS uses the current default disk drive.

path is the name of the DOS directory containing the FORMAT.COM file. If you omit the directory specifier, DOS uses the current default directory.

d: specifies the disk drive containing the disk to format.

/n:*sectors* defines the number of sectors per track. This is a DOS version 3.3 qualifier.

/S directs FORMAT to place the DOS system files on the disk, making the disk bootable.

/t:*tracks* defines the number of tracks per side. This is a DOS version 3.3 qualifier.

/V directs FORMAT to include the volume label.

/4 directs FORMAT to format the disk double-sided in a quad-density disk drive.

/1 directs FORMAT to format the disk as a single-sided disk.

/8 directs FORMAT to format the disk with eight sectors per track. Most disks use 9 or 15.

/B directs FORMAT to reserve space for the system files on the target disk. Unlike the /S option, /B does not actually place the files on disk.

NOTES The original manufacturer of disks you purchase has no way of knowing what computer the disks will be used on, nor with what operating system. Before you can use a new disk, therefore, you must format it under DOS.

The FORMAT /S command copies the DOS hidden files that are required to boot DOS to the target disk.

Because inadvertently formatting a fixed disk can be disastrous, FORMAT first prompts you with

```
WARNING, ALL DATA ON NON-REMOVEABLE DISK
DRIVE N: WILL BE LOST!
```

```
Proceed with Format (Y/N)?
```

To proceed with FORMAT, type **Y**; otherwise, type **N**.

Upon completion FORMAT displays

```
Total disk space
Corrupted disk space marked as defective
Total disk space consumed by the operating system
Total disk space available for file utilization
```

As you can see, FORMAT reports on defective space that it finds during formatting. In addition, FORMAT places entries for each defective sector into a table called the file allocation table, which prevents DOS from using the corrupted sectors for data storage.

Do not use FORMAT in conjunction with ASSIGN, JOIN, or SUBST.

FORMAT does not allow you to use the /S and /V qualifiers with the /B qualifier.

To support batch processing, FORMAT uses the following exit status values:

0	Successful format
3	User has terminated with CTRL-BREAK
4	Termination due to an error in processing
5	User has terminated, using N to fixed-disk prompt

FORMAT will not work with drives currently in a network configuration. The following are valid FORMAT qualifiers:

160/180K disks	/1 /4 /8 /B /N /T /V /S
320/360K disks	/1 /4 /8 /B /N /T /V /S
720K disks	/N /T /V /S
1.2MB disks	/N /T /V /S
1.44MB disks	/N /T /V /S
Fixed disks	/V /S

WARNING: FORMAT destroys any of the information contained on the target disk.

EXAMPLES The following command makes a bootable disk of the disk contained in drive A:

```
C> FORMAT A:/S
```

Many users often have to format double-density disks in their 1.2MB drives. The /4 qualifier in the FORMAT command directs FORMAT to create a 360K disk:

```
C> FORMAT A:/4
```

This command displays

```
Insert a new diskette in drive A:
and press Enter when ready.
```

MESSAGES If FORMAT displays the message

```
n bytes total disk space
n bytes used for system
n bytes in bad sectors
n bytes available on disk
```

it is providing you with a summary of the disk usage for the newly formatted disk. If FORMAT continually finds damaged sectors on your disks, you might consider a new brand of disk. With the state of floppy-disk technology today, such errors are much less common.

If FORMAT displays the message

```
Attempted write-protect violation
```

the disk in the target drive specified in your FORMAT command line is write protected. In most cases, this error occurs when you have forgotton to place an unused disk in the target disk drive. As you can see, write protecting a disk protects the information stored on the disk from errant DOS commands.

If FORMAT displays the message

```
Cannot find System Files
```

your FORMAT command line contains the /S qualifier, which directs FORMAT to create a bootable target disk, and FORMAT could not locate the DOS hidden system files that it must copy to the target disk in order for the disk to be bootable. Locate the disk that you used to boot DOS, and use it in the FORMAT operation.

If FORMAT displays the message

```
Cannot FORMAT a Network drive
```

the target disk drive that you have specified in the FORMAT command line is associated with a network drive. DOS will not allow you to format a network disk drive from a remote node in the network.

If FORMAT displays the message

```
Disk unsuitable for system disk
```

the specified disk contains damaged sectors in the first portion of the disk, which DOS used to store the hidden system files that start DOS. Since DOS cannot place these files on the disk at this location, FORMAT cannot create a bootable disk.

If FORMAT displays the message

```
Drive letter must be specified
```

your FORMAT command line does not contain a target disk drive letter. Reissue the FORMAT command, specifying the drive that will contain the disk to format.

If FORMAT displays the message

```
Drive not ready
```

either the target disk drive specified in the FORMAT command line does not contain a disk to format or the disk drive latch is not closed. Make sure that a disk is in the drive and close the drive latch. Repeat the FORMAT command.

If FORMAT displays the message

```
Enter current Volume Label for drive n:
```

FORMAT is prompting you to enter a disk volume name that it will assign to the disk upon completion of the format operation. Type the 11-character disk volume name, or press ENTER for no name. Remember, assigning meaningful names to your disks will improve your disk organization.

If FORMAT displays the message

```
Error in IOCTL call
```

it has encountered an internal error during the formatting process. Restart DOS and reissue the FORMAT command. In some cases, this error occurs when you specify a qualifier that is not supported by the target disk drive.

If FORMAT displays the message

```
Error reading partition table
```

it encountered an error while attempting to read the partition table that specifies your fixed disk's layout. Restart DOS and repeat the command. If this error persists, invoke the FDISK command to correct the partition table error.

If FORMAT displays the message

```
Error writing directory
```

it could not create a root directory on the specified disk. This error is normally associated with an unusable disk.

If FORMAT displays the message

```
Format another (Y/N)?
```

it has completed processing one disk (either successfully or unsuccessfully) and is prompting you to decide whether you want to format additional disks. If so,

type **Y** and press ENTER. FORMAT will prompt you to insert another disk. If not, type **N** and press ENTER. FORMAT will return control to DOS.

If FORMAT displays the message

```
Format complete
```

the format operation has successfully completed.

If FORMAT displays the message

```
Incorrect DOS version
```

the executable FORMAT.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct FORMAT.COM for the executing version of DOS.

If FORMAT displays the message

```
Insert DOS diskette for drive n
and strike ENTER when ready
```

your FORMAT command line contains the /S qualifier, which directs FORMAT to create a bootable DOS disk by copying the hidden DOS system files that DOS requires to boot. In this case, FORMAT is prompting you to insert a bootable DOS disk in the specified drive so that it can copy the hidden files to the target disk.

If FORMAT displays the message

```
Insert new diskette for drive n:
and strike ENTER when ready
```

it is prompting you to place the disk that you want to format in the specified drive. Once you do so, press ENTER and FORMAT will continue its processing.

If FORMAT displays the message

```
Invalid characters in volume label
```

your FORMAT command line contains the /V qualifier, which directs FORMAT to assign a disk volume name to the disk upon completion of the format operation. In this case, when FORMAT prompted you to enter a disk volume name,

one or more of the letters that you typed were invalid for a disk label name. Remember, volume names follow the same naming conventions as DOS files.

If FORMAT displays the message

```
Invalid drive specification
```

the target disk drive specified in your FORMAT command does not exist. Make sure that you have placed a colon after the disk drive letter and that you are using a valid drive letter (A, B, C, and so on). If your system has a fixed disk (drive C) and this error occurs, you have likely not used the FDISK command to tell DOS about your fixed disk.

If FORMAT displays the message

```
Bad Partition Table
```

the partition table that DOS uses to keep track of your fixed disk's layout is bad. Invoke the DOS FDISK command to correct your partition table layout.

If FORMAT displays the message

```
Invalid media or Track 0 bad - disk unusable
```

the first track on the specified disk is damaged. DOS must be able to format the first track of a disk either to make the disk bootable or to place information in the track that tells DOS the disk is not bootable. Therefore, if DOS cannot use this track, it cannot use the disk.

If FORMAT displays the message

```
Invalid parameter
```

one of the qualifiers in your FORMAT command line is invalid. Examine the FORMAT command format and reissue the command.

If FORMAT displays the message

```
Non-System disk or disk error
Replace and strike any key when ready
```

your FORMAT command line contains the /S qualifier, which directs FORMAT to create a bootable disk by copying the hidden DOS system files that DOS requires at startup to the disk. In this case, the disk in the specified drive did not contain the hidden system files, or FORMAT could not read the disk. Place the disk that you used to boot DOS in the drive and press any key to continue.

If FORMAT displays the message

```
Not a block device
```

the device specified in your FORMAT command line is not a block device (a disk or tape drive). Reissue the command with a valid disk device name. If you feel that your command is correct, make sure that you have installed a proper device driver for the device, if necessary.

If FORMAT displays the message

```
Parameters not compatible
```

two or more of the qualifiers in your FORMAT command line cannot be issued in conjunction with one another. Review the format of the FORMAT command and reissue the command.

If FORMAT displays the message

```
Parameters not compatible with fixed disk
```

one or more of the qualifiers in your FORMAT command line is invalid for a fixed-disk drive. Review the format of the FORMAT command and reissue the command.

If FORMAT displays the message

```
Parameters not supported by drive n:
```

one or more of the qualifiers in your FORMAT command line is invalid for the specified disk drive. Review the qualifiers as specified in the format of the FORMAT command and reissue the command.

If FORMAT displays the message

```
System transferred
```

your FORMAT command line contained the /S qualifier, which directs FORMAT to build a bootable disk and FORMAT has successfully copied the hidden system files to the target disk.

If FORMAT displays the message

```
Too many open files
```

it was unable to successfully create a volume label for the device or copy the hidden system files, because DOS already had too many files open. If this error occurs, make sure that you place the FILES=20 entry in CONFIG.SYS and reboot DOS. Once DOS restarts, repeat the FORMAT command.

If FORMAT displays the message

```
Volume label (11 characters, ENTER for none)?
```

your FORMAT command line contains the /V qualifier, which directs FORMAT to prompt you for a disk volume name that FORMAT will assign to the disk upon completion of the format operation. In this case, type the 11-character disk volume name, or press ENTER for no name. Remember, assigning meaningful names to your disks will improve your disk organization.

If FORMAT displays the message

```
Unable to write BOOT
```

it could not write the information to the first track of the disk, which FORMAT uses to create a bootable disk or to tell DOS that the disk is not a bootable disk. Repeat the FORMAT command to see if the error corrects itself. If not, the disk is unusable.

If FORMAT displays the message

```
WARNING, ALL DATA ON NON-REMOVABLE
DRIVE n: WILL BE LOST
Proceed with Format (Y/N)?
```

it is warning you that the disk specified in your FORMAT command line is a fixed disk, and that continuing with the FORMAT operation will destroy all of the information that the disk contains. If you want to proceed with the format

operation (overwriting the information on your fixed disk), type **Y** and press ENTER. Otherwise, type **N**, and press ENTER.

GOTO

FUNCTION Branches to the label specified in a BAT file.

COMMAND TYPE Internal

FORMAT

GOTO *label_name*

where *label_name* specifies the name of a label within a DOS batch procedure.

NOTES DOS label names can contain any of the characters that are valid for DOS file names. If the label does not exist, DOS terminates execution of the batch file.

DOS label names can be virtually any length; however, DOS distinguishes only the first eight characters. Therefore, DOS considers the label names DOS_LABEL1 and DOS_LABEL2 as equivalent.

EXAMPLES The following batch procedure displays the message "LOOPING" until the user presses CTRL-C or CTRL-BREAK.

```
ECHO OFF
:LOOP
ECHO LOOPING
GOTO LOOP
```

When this procedure is invoked, it repeatedly displays a directory listing of the current drive.

When DOS cannot find a label specified in a GOTO command, it terminates the processing, as shown here:

```
GOTO DOSLABEL
DATE
TIME
:DOSLABL
```

Upon invocation the procedure displays

```
Label not found
```

The following procedure illustrates that DOS examines only the first eight characters of a label name. Although the GOTO command branches to the label DOS_LABEL2, DOS first finds DOS_LABEL:

```
GOTO DOS_LABEL2
:DOS_LABEL
ECHO LABEL1
GOTO DONE
:DOS_LABEL2
ECHO LABEL2
:DONE
```

MESSAGES If GOTO displays the message

```
Label not found
```

the label following the GOTO command was not found in the batch file. When this occurs, DOS will terminate the batch-file processing.

GRAFTABL

FUNCTION Enables display of the extended character set when the display is in graphics mode.

COMMAND TYPE External GRAFTABL.COM

FORMAT

[*drive:*][*path*]GRAFTABL [*codepage* | /STATUS]

where the following is true:

drive: specifies the disk drive containing the GRAFTABL.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the DOS path name of the directory that contains the GRAFTABL.COM file. If you do not specify a DOS path name, DOS uses the current default directory.

codepage specifies one of the following codepages to be used for display:

437	United States
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

/STATUS directs GRAFTABL to display the codepage that is currently in use.

NOTES The DOS GRAFTABL command allows users to display extended ASCII characters when the display is in medium-resolution graphics mode.

GRAFTABL loads memory-resident code when it is invoked; therefore, it can be invoked only once.

To support your batch processing, GRAFTABL provides the following exit status values:

0	Successful
1	Graphics table already loaded

2 File error occurred during load

3 Invalid parameter specified, load not performed

4 Incorrect DOS version

EXAMPLES If you specify the /STA qualifier in the GRAFTABL command line

```
C> GRAFTABL /STA
```

GRAFTABL displays the number of the current codepage, as shown here:

```
USA version of Graphic Character Set Table is already loaded.
```

The following command directs GRAFTABL to use the codepage for the United States when it displays extended characters:

```
C> GRAFTABL 437
```

MESSAGES If GRAFTABL displays the message

```
xxxx version of Graphic Character Set Table is already loaded
```

you are attempting to invoke GRAFTABL a second time. GRAFTABL loads memory-resident software that aids DOS in printscreen operations. Therefore, you need only invoke GRAFTABL once per user session.

If GRAFTABL displays the message

```
DOS 2.0 or later required
```

you are invoking GRAFTABL from preversion 2 DOS. DOS did not provide support for GRAFTABL prior to version 2.

If GRAFTABL displays the message

```
Incorrect DOS version
```

the executable GRAFTABL.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct GRAPHICS.COM for the executing version of DOS.

If GRAFTABL displays the message

```
Invalid parameter
```

make sure that you have specified a codepage, a question mark, or the /STATUS qualifier in the GRAFTABL command line.

GRAPHICS

FUNCTION Allows screen contents containing graphics to be printed by means of printscreen operations.

COMMAND TYPE External GRAPHICS.COM

FORMAT

[*drive:*][*path*]GRAPHICS [*printer_type*][/B][/R][/P:*port_id*][/LCD]

where the following is true:

drive: specifies the disk drive that contains the GRAPHICS.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the complete DOS path name of the subdirectory that contains the GRAPHICS.COM file. If you do not specify a path name, DOS uses the current default directory.

printer_type specifies one of the following target printer types:

COLOR1	Color printer with black ribbon
COLOR4	Color printer with RGB ribbon

COLOR8	Color printer with cyan, magenta, yellow, and black ribbon
COMPACT	Compact printer
GRAPHICS	Graphics printer
THERMAL	Thermal printer

/B directs GRAPHICS to print the background color. The default is not to print the background color.

/LCD directs GRAPHICS to print the display from the PC Portable.

/P=*port_id* specifies the parallel printer port that GRAPHICS is to use. Valid values include 1, 2, or 3. The default is 1.

/R directs GRAPHICS to reverse the color of the screen image. Black images on the screen will be printed as white and white images as black.

NOTES GRAPHICS is a memory-resident program; therefore, you must install it only once.

EXAMPLE The following command loads the memory-resident software required to support printscreen operations that contain graphics images:

```
C> GRAPHICS
```

MESSAGES If GRAPHICS displays the message

```
DOS 2.0 or later required
```

you are invoking GRAPHICS from preversion 2 DOS. DOS did not provide support for GRAPHICS prior to version 2.

If GRAPHICS displays the message

```
Incorrect DOS version
```

the executable GRAPHICS.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct GRAPHICS.COM for the executing version of DOS.

If GRAPHICS displays the message

```
Unrecognized printer
```

the printer type that you have specified in the command is invalid. Make sure that your printer type is COLOR1, COLOR4, COLOR8, COMPACT, GRAPHICS, or THERMAL.

If GRAPHICS displays the message

```
Unrecognized printer port
```

the port that you specified for support in the command line is invalid. Make sure that you specify 1, 2, or 3 for LPT1, LPT2, or LPT3, respectively.

If GRAPHICS displays the message

```
Invalid parameter
```

one of the qualifiers in your GRAPHICS command line is invalid. Make sure that each qualifier is valid for your hardware configuration.

IF

FUNCTION Provides conditional processing within DOS batch files.

COMMAND TYPE Internal

FORMAT

IF [NOT] *condition DOS_command*

where the following is true:

NOT performs a Boolean NOT on the result of the condition.

condition must be one of the following:

- ERRORLEVEL value (true if program exit status >= value)
- EXIST file_spec (true if the specified file exists)
- string1==string2 (true if both strings are identical)

DOS_command is the name of the command that DOS is to perform if the condition is true.

NOTES Although the DOS IF statement is most commonly used from within DOS batch files, DOS fully supports IF from the command line, as shown here:

```
C> IF EXIST CONFIG.SYS TYPE CONFIG.SYS
```

EXAMPLES In the following command, if the CONFIG.SYS file exists in the current directory, DOS copies the file to drive B:

```
IF EXIST CONFIG.SYS COPY CONFIG.SYS B:
```

If the DOSPGM program exits with a status greater than or equal to 3, this command directs DOS to display the message "T H R E E":

```
ECHO OFF
DOSPGM
IF ERRORLEVEL 3 ECHO T H R E E
```

You can add the Boolean NOT to this expression to direct the program to terminate if the exit status is less than three, as shown here:

```
ECHO OFF
DOSPGM
IF NOT ERRORLEVEL 3 GOTO DONE
ECHO T H R E E
:DONE
```

In this command, the DOS IF command it used to determine whether or not the value of the batch parameter is NULL:

```
IF '%1' == '' GOTO NULL
```

If you use this expression within the following batch file, you can echo each of the batch-file parameters to the screen:

```
ECHO OFF
:LOOP
SHIFT
IF '%0'== '' GOTO DONE
ECHO %0
GOTO LOOP
:DONE
```

When you invoke the procedure with

```
C> ECHOTEST A B C D
```

it displays

```
C> ECHOTEST A B C D
C> ECHO OFF
A
B
C
D
```

MESSAGES If IF displays the message

```
Bad command or filename
```

the DOS command in your command line after the condition does not exist as specified.

If IF displays the message

```
Syntax error
```

the format of your command is invalid, which is very likely yielding an invalid condition.

JOIN

FUNCTION Joins a disk drive to a DOS path.

COMMAND TYPE External JOIN.EXE

FORMAT

```
[drive:][path]JOIN [d1: [d2:path]][/D]
```

where the following is true:

drive: is the disk drive identification of the disk that contains the JOIN.EXE file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the JOIN.EXE file. If you do not specify a disk drive, DOS uses the current default directory.

d1: specifies the disk drive to join to the path provided.

d2:path specifies the join directory.

/D directs JOIN to disconnect a previously joined disk.

NOTES JOIN makes two disks appear as one by joining a disk to a DOS path.

If you issue a JOIN command without any parameters, JOIN displays the current joins.

DOS joins a disk only to an empty DOS directory.

Do not use JOIN in conjunction with the following commands:

CHKDSK
DISKCOPY
FDISK
FORMAT
LABEL
RECOVER
SYS

EXAMPLES Before you can JOIN a disk to a directory, you must use MKDIR to create an empty directory:

```
C> MKDIR \JOINDIR
```

Use JOIN to connect a disk to the directory, as shown here:

```
C> JOIN B: \JOINDIR
```

References to C:\JOINDIR are identical to references to drive B. If drive B contains DOS directories, refer to them as

```
C> DIR \JOINDIR\SUBDIR
```

If you invoke JOIN without any command-line parameters,

```
C> JOIN
```

JOIN displays the current joins, as shown here:

To disconnect a join, use the /D qualifier, as shown here:

```
C> JOIN A: /D
```

MESSAGES If JOIN displays the message

```
Directory not empty
```

you are attempting to join a disk to a DOS subdirectory that contains other subdirectories or files. DOS requires that your join directory is empty prior to the JOIN command.

If JOIN displays the message

```
Not enough memory
```

DOS does not have enough available memory to support the JOIN command. You may need to remove memory-resident software or a RAM drive.

If JOIN displays the message

```
Incorrect number of parameters
```

you have either specified too few or too many command-line parameters. Examine the JOIN format and reissue the command.

If JOIN displays the message

```
Invalid parameter
```

you may not join a disk to the root directory.

If JOIN displays the message

```
Incorrect DOS version
```

the executable JOIN.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct JOIN.EXE for the executing version of DOS.

If JOIN displays the message

Cannot JOIN to a Network drive

you are trying to join a drive currently in the network to another device. DOS does not support this function.

KEYB

FUNCTION Loads a foreign keyboard set.

COMMAND TYPE External KEYB.COM

FORMAT

[*drive:*][*path*]KEYB [*keyboard_code* [,*codepage*],[*filespec*]]]

where the following is true:

keyboard_code specifies one of the following two-letter codes associated with the desired keyboard:

US	United States	FR	France
GR	Germany	IT	Italy
SP	Spain	UK	United Kingdom
PO	Portugal	SG	Swiss German
DK	Denmark	BE	Belgium
NL	Netherlands	NO	Norway
LA	Latin America	SV	Sweden
SU	Finland		

codepage is the desired codepage (see CHCP for a list of valid values).

filespec is the name of the file containing the keyboard definitions, normally KEYBOARD.SYS.

NOTES To fully support international configurations, DOS provides support for various keyboard templates.

KEYB loads memory-resident software to replace the standard keyboard layout supported by the ROM-BIOS.

When a new keyboard is installed, you can toggle between it and the default keyboard by pressing CTRL-ALT-F1 for the default, and CTRL-ALT-F2 for the foreign keyboard.

EXAMPLES Given the following command, DOS will use the United Kingdom keyboard template:

```
C> KEYB UK
```

This command selects the French keyboard template and informs DOS that the KEYBOARD.SYS file resides in the \DOS subdirectory:

```
C> KEYB FR,,\DOS\KEYBOARD.SYS
```

MESSAGES If, when you issue a KEYBxx command, DOS displays the message

```
Bad command or file name
```

the KEYBxx command that you are trying to execute does not reside on the current disk or you are using DOS version 3.3, which uses the KEYB command instead.

If, when you issue a KEYBxx command, DOS displays the message

```
Incorrect DOS version
```

the executable KEYBxx.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct KEYBxx.COM for the executing version of DOS.

If KEYB displays the message

```
Active Code Page not available from CON device
```

make sure that you have placed the DISPLAY.SYS entry in CONFIG.SYS and issued the correct MODE CON CODEPAGE PREPARE command for the codepage desired.

If KEYB displays the message

```
Bad or Missing Keyboard definition file
```

it cannot locate the KEYBOARD.SYS file. Make sure that KEYBOARD.SYS resides in the current directory, and that you specify a complete path name to the file in the KEYB command line.

If KEYB displays the message

```
Code page requested (nnn) is not valid
for given keyboard code
```

the command failed because your keyboard template and desired codepage are inconsistent. Specify a character codepage whose character set corresponds with the desired keyboard template.

If KEYB displays the message

```
Code page specified has not been prepared
```

make sure that you have specified a correct DISPLAY.SYS entry in CONFIG.SYS and that you issue the corresponding MODE CODEPAGE PREPARE command.

If KEYB displays the message

```
Current CON code page: nnn
```

it is telling you the active codepage (character set) for the CON device.

If KEYB displays the message

```
Current Keyboard code: nnn
code page: nnn
```

it is displaying the current keyboard template and codepage for the CON device.

If KEYB displays the message

```
Invalid code page specified
```

the 3-digit codepage that you have specified in the command line is invalid. Double-check your desired codepage entry number and reissue the command.

If KEYB displays the message

```
Invalid keyboard code specified
```

the 2-letter keyboard code that you have specified in the KEYB command line is invalid. Double-check your desired keyboard code and reissue the command.

If KEYB displays the message

```
Invalid syntax
```

you have very likely reversed the order of one of the KEYB command arguments. Review the format of the KEYB command and reissue the command.

If KEYB displays the message

```
KEYB has not been installed
```

it is telling you that due to an error (whose error message should also be displayed), the KEYB command failed and a keyboard template was not installed.

If KEYB displays the message

```
One or more CON code pages invalid for given
keyboard code
```

one of the 3-digit codepages that you have specified is invalid for the specified keyboard code. Review the codepage and keyboard code identifiers and reissue the command with the correct character set (codepage).

If KEYB displays the message

```
Unable to create KEYB table in
resident memory
```

your system has insufficient RAM to install the memory-resident portion of the KEYB command. If you have installed other memory-resident software or a RAM drive, try restarting DOS without them present in memory and reissuing the KEYB command.

LABEL

FUNCTION Specifies a disk volume label.

COMMAND TYPE External LABEL.COM

FORMAT

[*drive:*][*path*]LABEL [*target_drive:*][*volume_label*]

where the following is true:

drive: is the disk drive identification of the disk containing the LABEL.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the LABEL.COM file. If you do not specify a disk drive, DOS will use the current default directory.

target_drive: is the disk drive containing the disk to label.

volume_label is the 11-character volume label desired. All of the characters that are valid in DOS file names are valid volume label characters.

NOTES DOS allows you to define a name for each of your disks. Each time you issue the DOS DIR command it displays the volume label of the disk as well as its directory label, as shown here:

```
Volume in drive A is DOSLABEL
Directory of  A:\
```

It is also possible to use software to obtain the disk volume label from within your DOS programs. You can then be sure that the user has the correct disk in each drive.

The DOS VOL command also displays the disk volume label, as shown here:

```
C> VOL A:

Volume in drive A is DOSLABEL
```

If you do not specify a volume label in the command line, LABEL prompts you for one, as follows:

```
Volume in drive C is DOSDISK
Type a volume label of up to 11 characters
or press Enter for no volume label update.
```

If you do not want to change the disk label, press ENTER; otherwise, type the desired volume name.

Do not use the following characters in DOS volume names:

```
? \ / | . , ; : + = < > { } ( ) & ^
```

EXAMPLES The following command names the floppy disk contained in drive B as DOSDISK:

```
C> LABEL B:DOSDISK
```

The command line contains the label name, so LABEL did not have to prompt the user for any information.

In the command

```
C> LABEL
```

the label name is not specified in the command line; therefore, LABEL will prompt

```
Volume in drive C is DOSDISK
Volume label (11 characters, ENTER for none)?
```

Again, either type the volume label that you desire or press ENTER to leave the current label name unchanged.

MESSAGES If LABEL displays the message

```
Cannot LABEL a Network drive
```

the disk drive letter that you have specified in your LABEL command corresponds to a network disk drive. DOS will not allow you to assign a label to a drive in the network.

If LABEL displays the message

```
Delete current volume label (Y/N)?
```

you have pressed ENTER to the prompt to enter a new volume label and LABEL wants to know if you want to delete the current volume label. If so, type **Y** and press ENTER; otherwise, type **N**.

If LABEL displays the message

```
Incorrect DOS version
```

the executable LABEL.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct LABEL.COM for the executing version of DOS.

If LABEL displays the message

```
Invalid characters in volume label
```

one or more of the letters that you are using in the disk volume name is illegal. Remember, DOS label names follow the same naming conventions as your files.

If LABEL displays the message

```
No room in root directory
```

LABEL could not create the volume label because the root directory of the specified drive is full. DOS creates a hidden file in which it stores your disk volume label. If DOS cannot create this file, it cannot assign a name to your disk.

If LABEL displays the message

```
Too many files open
```

it could not open the hidden file that it will use to assign the name to your disk. If this error message occurs, increase the number specified in the FILES= entry in CONFIG.SYS.

If LABEL displays the message

```
Invalid drive specification
```

the disk drive that you have directed DOS to label is invalid. Make sure that you have included a colon after the disk drive letter and that you are using an existing disk drive (A, B, C, and so on).

If LABEL displays the message

```
Volume label (11 characters, ENTER for none)?
```

it is prompting you to type the name that you want to assign to the disk. Remember, DOS label names follow DOS file-naming conventions.

If LABEL displays the message

```
Volume in drive n is name
```

LABEL is telling you the current name for the disk in the specified drive.

MKDIR

FUNCTION Creates the specified DOS subdirectory.

COMMAND TYPE Internal

FORMAT

MKDIR [*drive:*]*path*

or

MD [*drive:*]*path*

where the following is true:

drive: specifies the drive on which to create the directory. If a drive is not specified, MKDIR uses the current default disk drive.

path specifies the name of the DOS directory that MKDIR is to create.

NOTES If you do not use DOS directories, your disks are restricted to a limited number of files, as shown here:

Disk Space	Maximum Number of Subdirectories in the Root Directory
160K	64
180K	
320K	112
360K	
1.2M	224
Fixed Disk	Based upon partition size

Every DOS directory has a root directory from within which all other directories grow.

The maximum path name that DOS can process is 63 characters.

MKDIR has two choices when creating a DOS directory. If the directory name starts with a backslash, such as \SUBDIR, DOS will start with the root directory to create the directory. If the name does not start with a backslash, as in SUBDIR, DOS will create the directory within the current directory.

Use the following guidelines when you create your DOS directories:

- DOS directory names conform to the same format as DOS file names—an 8-character name followed by an optional 3-character extension. Examples of valid DOS directory names are FILENAME.EXT, HARDWARE.SAL, and SOFTWARE.INV.
- If you do not specify a complete DOS path name when you create a directory, DOS assumes that you are creating the directory in the current directory.
- To manipulate directories contained on other disks, precede the directory name with a disk drive identification, such as B:\FINANCE\CAR.
- Do not create directory names that are identical to names of files contained in the same directory.
- Do not create a directory called \DEV. DOS uses a hidden directory called \DEV to communicate with hardware devices.
- DOS path names cannot exceed 63 characters.

- Root directories on each disk are restricted to a specific number of files due to the disk layout. Directories that you create, however, can contain an unlimited number of files.
- Logically divide your disk into directories.

EXAMPLES In the following example, MKDIR creates a directory called IBM in the root:

```
C> MKDIR \IBM
```

The following command creates a directory called NOTES in the IBM directory:

```
C> MKDIR \IBM\NOTES
```

This command is equivalent to the commands C> CHDIR \IBM and C> MKDIR NOTES. Note that the second command does not have a slash in front of the directory name. If it did, MKDIR would create the directory in the root as opposed to the \IBM directory.

Assuming that the current directory is still the root, the following command also creates a directory off the root:

```
C> MKDIR MISC
```

The directory name does not contain a slash, so MKDIR creates the directory in the current directory, which in this case is still the root. Had the current directory been other than the root, the correct command would have been

```
C> MKDIR \MISC
```

MESSAGES If MKDIR displays the message

```
Invalid drive specification
```

you have specified an invalid disk drive in the MKDIR command line. Make sure that you have included a colon following the disk drive letter and that you are using a valid disk drive letter (A, B, C, and so on).

If MKDIR displays the message

```
Invalid number of parameters
```

make sure that your command line contains a subdirectory name to create.

If MKDIR displays the message

```
Unable to create directory
```

one of the subdirectories in your directory path may be invalid, a subdirectory or file with the same name as the directory already exists on disk, or the root directory of the target disk drive is full.

MODE

FUNCTION Specifies device characteristics.

COMMAND TYPE External MODE.COM

FORMAT

[*drive:*][*path*]MODE *n*

or

[*drive:*][*path*]MODE [*n*],*m*,[*t*]

or

[*drive:*][*path*]MODE COM# [:] *baud* [,*parity*[,*data* [,*stop* [,P]]]]

or

[*drive:*][*path*]MODE LPT# [:] [*cpl*][,*vli*][,P]

or

[*drive:*][*path*]MODE LPT# [:]=COM# [:]

or

[*drive:*][*path*]MODE *device* CODEPAGE *operation*

where the following is true:

drive: specifies the disk drive identification of the disk containing the MODE.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path specifies the name of the DOS directory that contains the MODE.COM file. If you do not specify a directory name, DOS uses the current default directory.

n specifies the screen display attribute. It must be one of the following:

40	Specifies a 40-column displa
80	Specifies an 80-column display
BW40	Specifies a black-and-white 40-column display
BW80	Specifies a black-and-white 80-column display
CO40	Specifies a color 40-column display
CO80	Specifies a color 80-column display
MONO	Specifies a monochrome display

m specifies the direction to shift the screen display, one character to either the left or right.

t tells MODE to display a test pattern to aid in character alignment.

baud specifies the device baud rate: 110, 150, 300, 600, 1200, 2400, 4800, 9600, or 19200. MODE requires you to specify only the first two digits of the baud rate.

parity specifies the device parity: E for even parity, N for no parity, O for odd parity. The default is even parity.

data specifies the number of data bits, either 7 or 8. The default is 7 data bits.

stop specifies the number of stop bits, either 1 or 2. For 110 baud, the default is 2; otherwise, it is 1.

cpl is characters per line, either 80 or 132.

vli is vertical lines per inch, either 6 or 8.

P specifies continuous retries on timeout errors.

LPT# specifies the parallel printer number, such as LPT1.

COM# specifies the serial port number, such as COM1.

device specifies the device for which to manipulate a codepage, either CON or LPT*n*.

operation is the codepage operation desired:

PREPARE=[[*codepage*][*filename*]]

SELECT=*codepage*

REFRESH

/STATUS

NOTES Many hardware devices require unique data communications setups, such as 4800 baud and parity. The DOS MODE command allows you to set the characteristics of a port on the PC.

By default, DOS uses the parallel printer for printed data. If your printer is connected to a serial device, you can redirect the parallel output to the serial device by using MODE.

MESSAGES The following error messages are common to this command. Use this list of messages to resolve errors you experience when you issue commands from the DOS prompt.

If MODE displays the message

```
Incorrect DOS version
```

the executable MODE.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct MODE.COM for the executing version of DOS.

If MODE displays the message

```
Infinite retry of parallel printer timeout
```

you have specified the ,P parameter in the MODE command line to enable infinite retries if your printer has a timeout condition. When continuous retry is enabled, your computer will continually attempt to access your printer device or until the user presses CTRL-BREAK to return control back to DOS.

If MODE displays the message

```
No retry on parallel printer timeout
```

your MODE LPTn command line did not specify the ,P parameter, which directs DOS to perform continuous retries to access the printer device. This is the default setting. Most users will not have a need to enable continuous retries on printer operations.

If MODE displays the message

```
Invalid parameter
```

one of the qualifiers in your MODE command line is invalid. Examine the format of the MODE command desired and reissue the command.

If MODE displays the message

```
Invalid number of parameters
```

your MODE command line contains either too many or too few parameters. Examine the format of the MODE command and reissue the command.

If MODE displays the message

```
Illegal device name
```

one of the device names specified in your MODE command is invalid. Make sure that you are specifying either LPT1 through LPT3 or COM1 through COM4 and reissue the MODE command.

If MODE displays the message

```
LPTn: set for 80
```

MODE is simply telling you that it has set your printer to 80 characters per line. Depending upon your printer type, MODE may not actually be able to set your printer as specified. Regardless, MODE will still display the message shown above.

If MODE displays the message

```
LPTn: set for 132
```

MODE is simply telling you that it has set your printer to 132 characters per line. Depending upon your printer type, MODE may not actually be able to set your printer as specified. Regardless, MODE will still display the previos message.

If MODE displays the message

```
Printer error
```

MODE could not modify your printer as specified in the MODE command line.

If MODE displays the message

```
Printer lines per inch set
```

has set the number of lines of text per inch on your printer to either 6 or 8 lines per inch as specified in the MODE command line. Depending upon your printer type, MODE may not be able to set your printer as specified in the MODE command line. Regardless, MODE will display the previous message.

If MODE displays the message

```
Resident portion of MODE loaded
```

MODE has installed memory-resident software to handle either continuous retry on printer operations or a right or left shift of characters on your screen display. This message is purely informational.

If MODE displays the message

```
Do you see leftmost 0? (Y/N)
```

your MODE command line specified directed MODE to shift the characters on your screen display to the right. In this case, if you respond with **N** to the prompt, MODE will shift the characters to the left as desired. If you type **Y**, MODE will leave your screen unchanged. Most screen displays will never require this functionality.

If MODE displays the message

```
Do you see rightmost 9? (Y/N)
```

your MODE command line specified directed MODE to shift the characters on your screen display to the left. In this case, if you respond with **N** to the prompt, MODE will shift the characters to the right as desired. If you type **Y**, MODE will leave your screen unchanged. Most screen displays will never require this functionality.

If MODE displays the message

```
Requested screen shift out of range
Unable to shift Screen
```

it cannot shift the screen display to the left or right as specified in your MODE command. Some monitors do not support character shifting. Most monitors will never require this functionality.

If MODE displays the message

```
COMn: baud, parity, data bits, stop bits, time out
```

MODE is displaying the current data communication parameters for the communication port specified. Use this message to determine your COM port settings.

If MODE displays the message

```
COM port does not exist
```

the communications port specified in your MODE command line does not exist. Make sure that you are specifying COM1, COM2, COM3, or COM4 and reissue the MODE command.

If MODE displays the message

```
Invalid baud rate specified
```

your MODE COMn command line specifies an invalid baud rate. Use 110, 150, 300, 600, 1200, 2400, 4800, 9600, or 19200. Remember, MODE only requires you to specify the first two digits of the baud rate.

If MODE displays the message

```
LPTn: redirected to COMn:
```

your MODE command line has redirected output that DOS normally routes to the parallel port LPTn to the specified serial communications port. If you are using a serial printer, you will probably want to redirect output to your serial port in this manner.

If MODE displays the message

```
LPTn: not redirected
```

your MODE command line has set DOS printer output back to the parallel port specified following a previous MODE command that redirected printer output from a parallel device to a serial device.

If MODE displays the message

```
Active code page for device nnn is nnn
```

your MODE command line specified the CODEPAGE /STATUS option which directs MODE to display the current codepage for the specified device. This message is purely informational.

If MODE displays the message

```
Code page not prepared
```

your MODE command either includes the CODEPAGE /STATUS option which directs MODE to display the current codepage, or CODEPAGE SELECT which directs MODE to select a codepage for the specified device. If your command line contains /STATUS, this message is purely informational. If instead, you issued a CODEPAGE SELECT command, you must first issue the CODEPAGE PREPARE option with the MODE command.

If MODE displays the message

```
Code pages cannot be prepared
```

your MODE command contains a CODEPAGE PREPARE option that MODE cannot complete. Make sure that you have installed the DISPLAY.SYS and PRINTER.SYS device drivers in your CONFIG.SYS file.

If MODE displays the message

```
Code page operation not supported on this device
```

either the device name that you have specified in the MODE command is invalid or you have not installed the DISPLAY.SYS or PRINTER.SYS device driver in CONFIG.SYS.

If MODE displays the message

```
Device error during operation
```

the MODE command could not successfully complete due to a variety of possible errors. Make sure that you have installed the DISPLAY.SYS or PRINTER.SYS device driver in CONFIG.SYS, that the device is turned on, and that you are using a valid code page information file.

If MODE displays the message

```
Device or code page missing from font file
```

your MODE command line contains a CODEPAGE PREPARE option that MODE cannot complete. Make sure that your command line specifies a valid codepage number and that the device name is correct.

If MODE displays the message

```
Failure to access code page font file
```

your CODEPAGE PREPARE command failed because MODE could not locate the codepage information file specified. Make sure that the codepage information file specified resides in the current directory or contains a complete subdirectory path.

If MODE displays the message

```
Font file contents invalid
```

your CODEPAGE PREPARE command failed because MODE could not read the contents of the codepage information file. Reissue the command with the orginal DOS codepage information file.

If MODE displays the message

```
Must specify COM1, COM2, COM3, or COM4
```

the serial communications port that you have specified in your MODE command line is invalid. MODE is telling you to specify one of the communications ports specified.

If MODE displays the message

```
Failure to access device
```

MODE could not access the device specified in your MODE command line. Make sure that you have correctly specified the device name and that you have correctly installed any required device drivers in CONFIG.SYS.

If MODE displays the message

```
Previously prepared code page replaced
```

your MODE command has prepared a new codepage for the specified device, overwriting the previously prepared codepages. This message is informational.

If MODE displays the message

```
Device nnnn not prepared
```

your MODE COEPAGE PREPARE failed. Make sure that you have correctly specified the device name and installed the device drivers DISPLAY.SYS and PRINTER.SYS in CONFIG.SYS.

MORE

FUNCTION Displays a command's output one screenful at a time.

COMMAND TYPE External MORE.COM

FORMAT

DOS_command | [*drive:*][*path*]MORE

or

[*drive:*][*path*]MORE < *DOS_command*

where the following is true:

drive: is the disk drive identification of the disk containing the MORE.COM file. If you omit the disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the MORE.COM file. If you omit the directory name, DOS uses the current default directory.

NOTES The DOS MORE command reads data from the standard input device and displays the information to the standard output device one page at a time until the end of the file. Each time a page of data is displayed on the screen, MORE displays the message:

```
-- MORE --
```

Press any key to continue the output, or press CTRL-C to terminate the command.

EXAMPLES In the following command, MORE is used as a filter, obtaining its input from the standard input device (stdin):

```
C> SORT < DATA.DAT | MORE
```

To simply display the contents of BIGFILE.DAT one screenful at a time, use MORE as shown here:

```
C> MORE < BIGFILE.DAT
```

When you issue the MORE command from the DOS prompt, MORE expects its input from the keyboard.

```
C> MORE
```

Type a screenful of information and then press CTRL-Z. MORE will process as normal.

MESSAGES If MORE displays the message

```
MORE: Incorrect DOS version
```

the executable MORE.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct MORE.COM for the executing version of DOS.

NLSFUNC

FUNCTION Provides device support for international codepages.

COMMAND TYPE External NLSFUNC.EXE

FORMAT

[*drive:*][*path*]NLSFUNC [*file_spec*]

where the following is true:

drive: is the disk drive identification of the disk containing the NLSFUNC.COM file. If you omit the disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the NLSFUNC.COM file. If you omit the directory name, DOS uses the current default directory.

file_spec is the complete DOS file specification for the file containing the country information. This file is usually COUNTRY.SYS.

NOTES The NLSFUNC command works in conjunction with CHCP. You must invoke NLSFUNC before you invoke CHCP.

This is a DOS version 3.3 command.

The COUNTRY.SYS file contains the country-specific information.

EXAMPLE The following command informs DOS that the country information file (COUNTRY.SYS) resides in the directory \SYSFILES on drive C:

```
C> NLSFUNC C:\DOS\COUNTRY.SYS
```

MESSAGES If NLSFUNC displays the message

```
Incorrect DOS version
```

the executable NLSFUNC.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct NLSFUNC.EXE for the executing version of DOS.

If NLSFUNC displays the message

```
File not found
```

NLSFUNC could not find the COUNTRY.SYS file. If COUNTRY.SYS is not in the current directory, make sure that you specify the location of this file in the NLSFUNC command line.

If NLSFUNC displays the message

```
NLSFUNC already installed
```

you have already invoked NLSFUNC. NLSFUNC installs memory-resident software, which requires you to invoke the command only once per user session.

PATH

FUNCTION Defines the command-file search path that DOS uses each time it fails to locate a command as an internal command or within the current directory or the specified directory.

COMMAND TYPE Internal

FORMAT

PATH [*drive:*][*path* [;[*drive:*][*path*]...]

or

PATH ;

where the following is true:

drive: specifies a disk drive that DOS is to include in the command-file search path.

path specifies a DOS directory to include in the command-file search path.

... indicates that the disk drive and directories may be specified several times.

NOTES Each time DOS cannot find a command as an internal command or as an EXE, COM, BAT, or CMD file in the current directory, it searches to see if the user has defined a command-file search path.

The DOS PATH command allows you to define disk drives and directories to be included in the command-file search path.

EXAMPLES In the following command, if DOS cannot find the command, it searches the root directories on drives C, B, and A, in that order:

```
C> PATH C:\;B:\;A:\
```

The following PATH command directs DOS to search \DOS, \UTIL, and then \MISC, in that order:

```
C> PATH \DOS;\UTIL;\MISC
```

MESSAGES If PATH displays the message

```
No Path
```

there is currently no command-file search path defined. DOS will search for your batch or executable files only in the current subdirectory or specified subdirectory.

If PATH displays the message

```
Out of environment space
```

you have very likely installed a memory-resident program (GRAPHICS, PRINT, or a third-party product) that prevents DOS from expanding the environment to store your new entry. If necessary, restart DOS and issue your PATH command prior to invoking the memory-resident program. Remember, by using SHELL= in CONFIG.SYS, you can specify the desired size of your DOS environment.

PAUSE

FUNCTION Pauses batch-file execution to display an optional message.

COMMAND TYPE Internal

FORMAT

PAUSE [*message*]

where *message* is an optional message that PAUSE is to display each time it suspends batch processing. The message can contain up to 123 characters.

NOTES Each time DOS encounters PAUSE within a batch file, it displays the following:

```
[optional message text]
Strike a key when ready . . .
```

To continue batch processing, press any key; otherwise, press CTRL-BREAK. If you press CTRL-BREAK, DOS displays

```
Terminate batch job (Y/N)?
```

To terminate the batch file, type **Y**; otherwise, type **N**.

The DOS ECHO OFF command suppresses the display of messages from PAUSE.

EXAMPLES In the following command

```
PAUSE Make sure that your printer is on-line
```

when DOS encounters the PAUSE command within the batch procedure, it pauses and displays

```
C> PAUSE Make sure that your printer is on-line
Make sure that your printer is on-line
Strike a key when ready . . .
```

The command

```
PAUSE
```

displays

```
C> PAUSE
Strike a key when ready . . .
```

MESSAGES If PAUSE simply displays

```
Strike a key when ready . . .
```

without the message you have specified in the PAUSE command line, the current ECHO status is OFF. When ECHO is OFF, DOS suppresses the display of messages by PAUSE.

PREDEFINED FUNCTION KEYS

FUNCTION As follows:

NOTES

F1	Copies one character from the previous command buffer
F2	Copies all characters in the buffer that precede the next character typed
F3	Copies all characters in the command buffer
F4	Copies all characters including and following the next character typed
F5	Edits the current command buffer

F6	Places a CONTROL-Z (^Z) end-of-file marker at the end of a file
INS	Inserts characters in the current command buffer
DEL	Deletes the character that precedes the cursor
ESC	Cancels the current command line without executing it

Whenever you press one of the function keys, DOS buffers it in a location in memory that you can access to simplify your command entry.

PRINT

FUNCTION Prints a DOS file from the print queue.

COMMAND TYPE External PRINT.COM

FORMAT

[*drive:*][*path*]PRINT [/D:*device_name*][/C][/T]
[/B:*buffersize*][/D:*device*][/M:*maxticks*][/Q:*queuesize*][/S:*timeslice*][/U:*busyticks*]
file_spec [...]

where the following is true:

drive: is the disk drive identification of the disk that contains the PRINT.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the PRINT.COM file. If you do not specify a disk drive, DOS uses the current default directory.

/D:*device_name* specifies the name of the device DOS is to print, such as LPT1. The default device is LPT1.

/C directs PRINT to cancel the print job of the file whose name precedes the /C and all those that follow.

/T directs PRINT to cancel all print jobs in the printer queue.

/B:*buffersize* specifies the amount of memory (in bytes) that is set aside for PRINT. The default size is 512. Increasing this size in multiples of 512 (1024, 2048, 4096) decreases the number of disk-I/O operations required, and thus improves PRINT's performance. Increasing this value does, however, consume memory. This qualifier is only valid the first time you invoke PRINT.

/D:*device* specifies the printer device (such as LPT1, LPT2, or LPT3). This qualifier is valid only the first time you invoke PRINT.

/M:*maxticks* specifies the maximum number of CPU clock ticks that PRINT can consume before it must return control to DOS. This value ranges from 1 to 255, and the default value is 2. This qualifier is valid only the first time you invoke PRINT. Increasing this value improves PRINT's performance by giving it more control of the CPU. However, if you make this value too large, the rest of your applications become sluggish when PRINT is working.

/Q:*queuesize* specifies the number of entries that the PRINT queue can store. The value must be in the range from 1 to 32, and the default value is 10. This qualifier is valid only the first time you invoke PRINT.

/S:*timeslice* specifies the PRINT time slice. This value must be in the range from 1 to 255, and the default value is 8. This qualifier is valid only the first time that you invoke PRINT.

/U:*busyticks* specifies the number of CPU clock ticks that PRINT will wait for the printer to become available for the next series of characters. The value must be in the range from 1 to 255, and the default value is 1. This qualifier is valid only the first time that you invoke PRINT.

file_spec is the complete DOS path name of the file to be added to or removed from the print queue. PRINT supports DOS wildcard characters.

... indicates that several file names can be placed on the PRINT command line.

NOTES CPU clock ticks occur 18.2 times per second on the IBM PC.

PRINT sends files to the printer in background mode, which allows you to continue your processing in the foreground.

Do not remove a disk that contains a file to be printed until the printing process has finished. If you do so, your printed listing fills with unrecognizable characters, and you must reissue the PRINT command.

EXAMPLES The following command installs a PRINT queue with storage for 32 files:

```
C> PRINT /Q:32
```

Remember that many qualifiers are valid only when you first issue a PRINT command.

To print the TEST.DAT file that resides in the current directory, invoke the PRINT command as

```
C> PRINT TEST.DAT
```

If the file resides in the MISC subdirectory, use the command

```
C> PRINT \MISC\TEST.DAT
```

This command sends all of the files in the current directory that have the DAT extension:

```
C> PRINT *.DAT
```

This command prints the CONFIG.SYS file on drive B:

```
C> PRINT B:CONFIG.SYS
```

The following command terminates all of the current print jobs:

```
C> PRINT /T
```

This command removes the AUTOEXEC.BAT file from the print queue:

```
C> PRINT AUTOEXEC.BAT /C
```

MESSAGES If PRINT prints the message

```
FILENAME.EXT File not found
```

you have either deleted or renamed the file after placing it in the print queue, but before it had been printed. You must allow files in the print queue to print before you can remove them from disk or remove the floppy disk that contains the file.

If PRINT displays the message

```
FILENAME.EXT File not found in print queue
```

it could not find the specified file in a command that contains a /C qualifier, directing PRINT to remove the file from the print queue. Redisplay the files in the print queue and then reissue the command.

If PRINT displays the message

```
FILENAME.EXT is currently being printed
```

it is telling you the name of the file that it is currently printing.

If PRINT displays the message

```
FILENAME.EXT is in queue
```

it is telling you that the specified file is in the print queue waiting to be printed.

If PRINT displays the message

```
FILENAME.EXT Pathname too long
```

the complete path name specified exceeds 63 characters. DOS will not process paths longer than 63 characters.

If PRINT displays the message

```
Access denied
```

the specified file is currently locked by another application. Wait a few minutes and repeat the PRINT command.

If PRINT prints the message

```
All files canceled by operator
```

you have canceled the printing of all of the files in the print queue by using the /T command-line qualifier.

If PRINT displays the message

```
Errors on list device indicate that it
may be off-line. Please check it.
```

your printer may either have been turned off or placed off-line. Turn on your printer, or place it on-line, and printing will continue.

If PRINT prints the message

```
File FILENAME.EXT canceled by operator
```

the specified file was removed by a /C qualifier in the PRINT command line and will not be printed.

If PRINT displays the message

```
Incorrect DOS version
```

the executable PRINT.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct PRINT.EXE for the executing version of DOS.

If PRINT displays the message

```
Invalid drive specification
```

a disk drive letter in your PRINT command line is invalid. Recheck your command line to ensure that you have included a colon after the disk drive letter and that you are using a valid disk drive letter (A, B, C, and so on).

If PRINT displays the message

```
Invalid parameter
```

your command line includes a qualifier that is not supported by the PRINT command. Check the format of the PRINT command and reissue the command.

If PRINT displays the message

```
List output is not assigned to a device
```

the specified device name at PRINT's prompt for a target device is invalid. Reissue the command, specifying a device name such as PRN, LPT1, COM1, and so on.

If PRINT displays the message

```
Name of list device [PRN]:
```

it is prompting you to enter the name of the target device where you want printed output to be written. By default, PRINT uses the device PRN. If you simply press ENTER, PRINT will select this device. If your printer is attached to a different port, such as COM1, type that device name.

If PRINT displays the message

```
No paper error writing device nnnnn
```

your printer is out of paper. Take the printer off-line, add paper, and place the printer back on-line. PRINT will continue printing your output.

If PRINT displays the message

```
PRINT queue is empty
```

there are no files in your print queue either printing or waiting to be printed.

If PRINT displays the message

```
PRINT queue is full
```

your printer queue is full and cannot hold an additional file until it has printed another file. Remember, the /Q qualifier allows you to specify a queue size of up

to 32 entries. You can specify /Q only the first time that you invoke the PRINT command.

If PRINT displays the message

```
Resident part of PRINT installed
```

it has installed the memory-resident software that DOS requires to handle your files and control of the printer.

PROMPT

FUNCTION Defines the DOS prompt that appears on your screen display.

COMMAND TYPE Internal

FORMAT

PROMPT [*prompt_string*]

where *prompt_string* is the character string that defines the DOS prompt. It can contain characters or the following metastrings:

$b	| character
$d	Date
$e	ESC
$h	BACKSPACE
$g	> character
$l	< character
$n	Current drive
$p	Current directory

$q	= character
$t	Current time
$v	DOS version
$_	CR-LF
$$	$ character

NOTES If no string is specified, PROMPT resets the system prompt to the current default drive.

EXAMPLES The following command sets the user's prompt to YES>:

```
C> PROMPT YES$g
YES>
```

This command sets the system prompt to the current system time:

```
C> PROMPT $t
```

This command displays:

```
15:20:18.81
15:20:43.78
15:20:87.12
```

To display only the hours and minutes, you can change the prompt to

```
C> PROMPT $t$h$h$h$h$h
```

You can use the DOS prompt to help keep track of the current directory. The following PROMPT command directs DOS to display the current directory name as the system prompt:

```
C> PROMPT [$p]

[C:\SUBDIR]
```

MESSAGES If PROMPT displays the message

```
Out of environment space
```

the environment is full. Until you install memory-resident software such as PRINT, GRAPHICS, or a third-party product, DOS allows your environment to grow to 32K as your processing needs require. If, however, you install memory-resident software, your environment size becomes fixed. Try to invoke all of your SET commands prior to loading memory-resident software. Also, remember that the SHELL= entry in CONFIG.SYS allows you to specify the environment size.

RECOVER

FUNCTION Recovers a damaged disk or file.

COMMAND TYPE External RECOVER.COM

FORMAT

[*drive:*][*path*]RECOVER [*d:*][*p*]*filename.ext*

where the following is true:

drive: specifies the disk drive containing the RECOVER.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the DOS path name of the directory that contains the RECOVER.COM file. If you do not specify a DOS path name, DOS uses the current default directory.

d: is the disk drive identification of the file or disk to recover. If you do not specify a disk drive, RECOVER uses the current default disk drive.

p is the DOS path name of the directory containing the file to recover. If you do not specify a DOS path name, DOS uses the current default directory.

filename.ext is the name of the damaged file to recover.

NOTES If a DOS disk or file becomes damaged and loses sectors, the DOS RECOVER command allows you to retrieve portions of the file up to the point of the corruption.

If the file is a text file, you can later edit the file and restore the missing contents. If, however, the file is an executable file, you should not execute it—remember, the file is missing sectors. Maintain a good backup of your files so that you do not have to rely on RECOVER.

If you use RECOVER to recover a complete disk, RECOVER creates files in the root directory with names in the form FILE*nnnn*.REC where *nnnn* is a 4-digit number beginning with 0001 (FILE0001.REC).

RECOVER does not work with disk drives connected to a network.

EXAMPLES The following command attempts to recover the contents of the disk in drive A:

```
C> RECOVER A:
```

RECOVER creates several files whose names are in the format FILE*nnnn*.REC.

The command

```
C> RECOVER FILENAME.EXT
```

recovers the contents of the FILENAME.EXT file up to the point of the damaged sector.

MESSAGES If RECOVER displays the message

```
Cannot RECOVER a Network drive
```

the disk drive that you are specifying in the RECOVER command line is a network device. DOS does not allow you to use RECOVER on a network disk.

If RECOVER displays the message

```
File not found
```

the file name that you specified in the command does not exist. Recheck your spelling and specified directory path.

If RECOVER displays the message

```
Invalid number of parameters
```

you have quite probably specified more than one disk drive in the RECOVER command line.

If RECOVER displays the message

```
Invalid drive or file name
```

either the disk drive or file name that you specified in the RECOVER command line is invalid.

If RECOVER displays the message

```
Incorrect DOS version
```

the executable RECOVER.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct RECOVER.COM for the executing version of DOS.

If RECOVER displays the message

```
Warning - directory full n file(s) recovered
```

the root directory of your disk drive is full. Copy the files FILE000*n*.REC to a new disk, delete the current files, and repeat the RECOVER command to continue recovering files on the disk drive.

REM

FUNCTION Displays comments during batch-file execution.

COMMAND TYPE Internal

FORMAT

REM [*message*]

where *message* is a character string of up to 123 characters.

NOTES REM allows you to display messages to the standard output device during the execution of batch (BAT) files. *Message* is an optional command-line parameter that contains the message. The DOS command ECHO OFF inhibits the display of messages by REM.

EXAMPLE The following procedure shows the use of REM in a batch file:

```
REM ****************************************
REM DATETIME.BAT
REM
REM This batch file invokes the DOS DATE
REM command followed immediately by the DOS
REM TIME command.
REM
REM Written 7/23/88 by K. Jamsa
REM
```

RENAME

FUNCTION Renames the file(s) specified.

COMMAND TYPE Internal

FORMAT

REN *file_specification filename[.ext]*

or

RENAME *file_specification filename[.ext]*

where the following is true:

file_specification is the complete DOS path name of the file to rename. It can contain a drive and DOS directory path.

filename[.ext] is the target file name for the rename. It cannot have a drive or DOS directory path. RENAME supports DOS wildcard characters

NOTES The target file must reside in the same directory on the same disk drive as the source file, because RENAME does not copy file contents, but rather renames the file in its directory entry.

EXAMPLES The following command gives all of the files on drive B that have the extension BAK the same file name and a new extension of SAV:

```
C> REN B:*.BAK *.SAV
```

This RENAME command changes the extension of all of the files with the extension SYS that are contained in the \DOS directory to DOS:

```
C> RENAME \DOS\*.SYS *.DOS
```

The target file must reside on the same disk and directory as the source. If you specify a disk drive on the target file, for example, REN displays

```
Invalid parameter
```

REPLACE

FUNCTION Allows selective file replacements and updates when new versions of software become available.

COMMAND TYPE External REPLACE.EXE

FORMAT

[*drive:*][*path*]REPLACE *source_filespec* [*target_filespec*] [/A][/P][/R][/S][/W]

where the following is true:

drive: specifies the disk drive containing the REPLACE.EXE file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the DOS path name of the directory that contains the REPLACE.EXE file. If you do not specify a DOS path name, DOS uses the current default directory.

source_filespec is the complete DOS file specification of the files that REPLACE is to use in the file replacement. REPLACE supports DOS wildcard characters.

target_filespec is the complete DOS file specification of the destination of the files being added or replaced.

/A directs REPLACE to add files to the target directory instead of replacing them. With this qualifier, REPLACE places only those files onto the target that are not currently present.

/P directs REPLACE to prompt

```
Replace drive:filename.ext (Y/N)?
```

before adding or replacing files.

/R directs REPLACE to also replace files on the target location that are currently marked as read-only. Without this qualifier, REPLACE stops replacement operations with the first file marked read-only.

/S directs REPLACE to search the directories on the target location for other occurrences of the file to be replaced. This qualifier cannot be used with /A.

/W directs REPLACE to prompt

```
Press any key to begin replacing file(s).
```

before starting the file replacement operations.

NOTES REPLACE is a convenient utility for software developers. It allows you to easily select specific files for replacement.

To support your batch processing, REPLACE provides the following exit status values.

0	Successful
1	Invalid command line
2	File not found
3	Path not found
5	Access denied
8	Insufficient memory
15	Invalid drive

EXAMPLES The following REPLACE command replaces any files in the root directory on drive A that have the extension A:

```
C> REPLACE *.A A:\
```

The following command replaces all of the files in the directory \A on drive A that have the extension A:

```
C> REPLACE *.A A:\A
```

The command

```
C> REPLACE *.* A:\ /S
```

updates all of the files on the entire disk in drive A (even those in directories) that are found on the source disk.

MESSAGES If REPLACE displays the message

```
n File(s) added
```

your REPLACE command line contains the /A qualifier, which directs REPLACE to add files to the target disk and REPLACE is telling you the number of files that it has added to the disk.

If REPLACE displays the message

```
n File(s) replaced
```

it is telling you the number of files that it has replaced on the target disk.

If REPLACE displays the message

```
Access denied 'pathname'
```

it could not replace the specified file because it has been marked read-only. If you want REPLACE to replace a read-only file, use the /R qualifier.

If REPLACE displays the message

```
Add pathname? (Y/N)
```

you have invoked REPLACE with the /A and /P qualifiers to selectively add files to the target disk. In this case, to add the file to the disk, type **Y** and press ENTER. If you type **N** and press ENTER, REPLACE will not modify the file.

If REPLACE displays the message

```
Adding pathname
```

your REPLACE command line contains the /A qualifier, which directs REPLACE to add files found on the source disk to the files that reside on the target. In this case, REPLACE is telling you that it is adding the specified file to the target disk.

If REPLACE displays the message

```
File cannot be copied onto itself 'path'
```

a source file for your REPLACE command is identical to the target. Change the target file name and reissue the command, or add a disk drive or directory specifier as required.

If REPLACE displays the message

```
Incorrect DOS version
```

the executable REPLACE.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct REPLACE.EXE for the executing version of DOS.

If REPLACE displays the message

```
Insufficient disk space
```

the target disk does not contain adequate disk space for the REPLACE command to complete successfully. If you have installed memory-resident software or a RAM drive, consider restarting DOS without them and repeating the command.

If REPLACE displays the message

```
Invalid drive specification 'n:'
```

either your source or target disk drive is invalid. Make sure that you have included a colon after the disk drive and are using an existing disk drive (A, B, C, and so on).

If REPLACE displays the message

```
No files added
```

your REPLACE command line contains the /A qualifier, which directs REPLACE to only add those files from the source disk that do not currently reside on the target disk. In this case, REPLACE found no files to add to the disk.

If REPLACE displays the message

```
No files found 'path'
```

no files existed on the specified source path. Make sure that the specified directory is valid and contains files, repeating the command as necessary.

If REPLACE displays the message

```
No files found
```

it did not find any files on the target disk that needed to be replaced by files on the source.

If REPLACE displays the message

```
Parameters not compatible
```

two or more of the specified qualifiers in the REPLACE command line are incompatible. Review the format of the REPLACE command and reissue the REPLACE command.

If REPLACE displays the message

```
Path not found '\path'
```

it could not locate the specified directory path. Make sure that you are specifying a valid subdirectory starting with the root directory if required.

If REPLACE displays the message

```
Path too long
```

a subdirectory path name in your REPLACE command line exceeds 63 characters. DOS does not support subdirectory path names of this length.

If REPLACE displays the message

```
Press any key to begin adding file(s)
```

your REPLACE command contains the /A qualifier, which directs REPLACE to add files from the source disk to the target; and also /W, which directs REPLACE to pause, allowing you to insert floppy disks into the correct disk drives.

If REPLACE displays the message

```
Press any key to begin replacing file(s)
```

your REPLACE command line contains the /W qualifier, which directs REPLACE to pause until the user presses a key. This allows you to insert the required disks into the correct disk drives.

If REPLACE displays the message

```
Replace FILENAME.EXT? (Y/N)
```

your REPLACE command line contains the /P qualifier, which allows you to perform selective file replacements. To replace the specified file, type **Y** and press ENTER. If you type **N**, REPLACE will not update the specified file.

If REPLACE displays the message

```
Source path required
```

your REPLACE command line does not specify the location of the files to use in the replacement operation. You must specify the files you want REPLACE to use in the command line.

If REPLACE displays the message

File not found

could not locate the specified file in the REPLACE command line.

RESTORE

FUNCTION Restores files saved by BACKUP.

COMMAND TYPE External RESTORE.COM

FORMAT

[*drive:*][*path*]RESTORE *source_drive target_drive:file_spec*
[/P][/S][/B:*mm-dd-yy*][/A:*mm-dd-yy*][/E:*hh:mm:ss*][/L:*hh:mm:ss*][/M][/N]

where the following is true:

drive: is the disk drive identification of the disk that contains the RESTORE.COM file. If you omit the disk drive specifier, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the RESTORE.COM file. If you do not specify a path name, DOS will use the current default directory.

target_drive:file_spec is the disk drive identification to which the files will be restored.

source_drive specifies the files to restore. The file name must match the name of the file as it was originally backed up. *Source_drive* is the drive that contains the backup files.

/A:*mm-dd-yy* directs RESTORE to restore only files modified after the specified date.

/B:*mm-dd-yy* directs RESTORE to restore only files modified on or before the specified date.

/E:*hh:mm:ss* directs RESTORE to restore only files modified on or before the specified time. This is a DOS version 3.3 qualifier.

/L:*hh:mm:ss* directs RESTORE to restore only files modified later than the specified time. This is a DOS version 3.3 qualifier.

/M directs RESTORE to restore only files modified since the last backup.

/N directs RESTORE to restore only files that no longer exist on the target disk.

/P directs RESTORE to prompt the user before restoring files that have been modified or set to read-only since the backup.

/S directs RESTORE to restore files contained in subdirectories.

NOTES The DOS BACKUP command places files onto a disk in a manner only accessible by RESTORE. To copy a file from the backup disk, you must use RESTORE.

RESTORE does not restore the hidden system files or the DOS command processor COMMAND.COM.

To support DOS batch processing, RESTORE provides the following exit status values.

0	Successful restoration
1	No files found to restore
2	Shared files not restored
3	User-terminated using CTRL-C
4	Restoration error

EXAMPLES This command restores all of the files from the backup disk in drive A, including those in directories:

```
C> RESTORE A: C:*.* /S
```

If the backup used several floppy disks, RESTORE prompts you to place subsequent disks in the specified drive each time it needs a new backup disk.

This command restores all of the files from the backup disk that have the extension DAT:

```
C> RESTORE A: C:*.DAT /P
```

RESTORE prompts the user with

```
Warning! File FILENAME.EXT was changed after it was backed up.
Replace the file (Y/N)?
```

before it restores files that have been modified since the backup.

MESSAGES If RESTORE displays the message

```
*** Files were backed up at hh:mm:ss on mm:dd:yy ***
```

it is telling you the date and time when the files on the backup disk were backed up. If this date and time does not correspond to your last known BACKUP operation, you may not be using your latest backup disks.

If RESTORE displays the message

```
*** Not able to restore file ***
```

the specified file may be currently in use by another program in the network. It is a good practice to perform BACKUP and RESTORE operations only when the network is disabled.

If RESTORE displays the message

```
*** Restoring files from drive n ***
Diskette nnn
```

it is telling you the disk drive letter and the backup disk number that it is currently using to restore files.

If RESTORE displays the message

```
File creation error
```

```
File creation error
```

it could not restore the specified file. This error may occur when a subdirectory has been created with a name that matches the file, or when the root directory of the target disk is full.

If RESTORE displays the message

```
Incorrect DOS version
```

the executable RESTORE.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct RESTORE.COM for the executing version of DOS.

If RESTORE displays the message

```
Insert backup diskette n in drive n
Strike any key when ready
```

it is telling you that it is ready for the next backup disk. Once you have inserted the disk in the specified drive, press a key and RESTORE will continue its processing.

If RESTORE displays the message

```
Insufficient memory
```

your system does not have enough available RAM for DOS to execute the RESTORE command. If you have installed any memory-resident software or a RAM drive, consider restarting DOS without them so that you can execute the RESTORE command.

If RESTORE displays the message

```
Invalid drive specification
```

one of the disk drive specifiers in your RESTORE command line is invalid. Make sure that you have placed a colon after the disk drive letter and that you are using a valid disk drive letter (A, B, C, and so on). Remember, you must specify the disk drive containing the backup files first in the RESTORE command line.

If RESTORE displays the message

```
Invalid number of parameters
```

your RESTORE command line has either too few or too many arguments. Examine the format of the RESTORE command and reissue the command.

If RESTORE displays the message

```
Invalid parameter
```

one of the qualifiers in your RESTORE command line is invalid. Examine the RESTORE command format and reissue the RESTORE command.

If RESTORE displays the message

```
Invalid path
```

your RESTORE command line specifies an invalid DOS subdirectory. Make sure that you have correctly specified the subdirectory structure to restore and reissue the RESTORE command.

If RESTORE displays the message

```
Restore file sequence error
```

it is performing a multiple-disk file restoration and you have inserted a backup disk out of order. You may need to repeat the command, inserting the disks in the correct order.

If RESTORE displays the message

```
Source and target drives are the same
```

the disk drives that you have specified in your RESTORE command line are the same. RESTORE does not support this type of processing. Reissue the command with a different source and target disk drive.

If RESTORE displays the message

```
Source does not contain backup files
```

the disk in the specified source drive in the RESTORE command line does not contain backup files. Repeat the command, inserting a disk in the drive that contains files placed on the disk by the BACKUP command.

If RESTORE displays the message

```
System files restored
Target diskette may not be bootable
```

it is warning you that it has restored DOS hidden system files, which may render the disk unbootable. If your disk no longer boots, you may need to use the DOS SYS command to make the disk bootable.

If RESTORE displays the message

```
Target is full
```

the target disk cannot store additional files. The last file specified on your screen display may not have been copied to the disk.

If RESTORE displays the message

```
The last file was not restored
disk full or file bad
```

it did not restore the last file specified on the screen display because either the target disk is full or the backup file is bad. If you disk is full, you may need to remove unnecessary files from your disk before you can restore the files.

If RESTORE displays the message

```
Warning! Diskette is out of sequence
Replace diskette or continue if okay
```

you have inserted a backup disk in the wrong order. If you want RESTORE to continue with this disk, press ENTER. If you want to insert a different disk, do so and press ENTER.

If RESTORE displays the message

```
Warning! File FILENAME.EXT
is a hidden file
Replace the file (Y/N)?
```

your RESTORE command line contains a /P qualifier, which allows you to selectively restore hidden files. If you want to restore the specified file, type **Y** and press ENTER. If you do not want to restore the file, type **N** and press ENTER.

If RESTORE displays the message

```
Warning! File FILENAME.EXT
is a read-only file
Replace the file (Y/N)?
```

your RESTORE command line contains a /P qualifier, which allows you to selectively restore files marked as read-only. If you want to restore the specified file, type **Y** and press ENTER. If you do not want to restore the file, type **N** and press ENTER.

If RESTORE displays the message

```
Warning! File FILENAME.EXT
changed after it was backed up
Replace the file (Y/N)?
```

your RESTORE command line contains a /P qualifier, which allows you to selectively restore files that have changed since the back up operaation was performed. If you want to restore the specified file, type **Y** and press ENTER. If you do not want to restore the file, type **N** and press ENTER.

If RESTORE displays the message

```
Warning! No files were found to restore
```

it did not find any matching files to restore from the backup disk. Make sure that you have correctly specified the desired files including the subdirectories in which they originally resided.

RMDIR

FUNCTION Removes the specified directory.

COMMAND TYPE Internal

FORMAT

RMDIR [*drive:*]*path*

or

RD [*drive:*]*path*

where the following is true:

drive: specifies the drive from which to remove the directory. If a drive is not specified, RMDIR uses the current default disk drive.

path specifies the name of the DOS subdirectory to remove.

NOTES RMDIR removes only empty directories that do not contain files. The largest path name that DOS can process is 63 characters.

EXAMPLES This command attempts to remove the IBM directory from the root directory of the current drive:

```
C> RMDIR \IBM
```

If the directory contains files, RMDIR cannot remove the directory.

The command

```
C> RMDIR MISC\IBM\SALES
```

removes the SALES directory from the \MISC\IBM directory on the current drive.

MESSAGES If RMDIR displays the message

```
Invalid path, not directory, or directory not empty
```

the directory name (or path name) that you are specifying is invalid, the specified name is not a DOS subdirectory, or the directory still contains files or subdirectories. DOS will not allow you to remove a subdirectory that contains files.

SELECT

FUNCTION Selects an international format for a new disk.

COMMAND TYPE External SELECT.COM

FORMAT

[*drive:*][*path*]SELECT [[A: | B:] *d:*[*p*]] *country keyboard*

where the following is true:

drive: is the disk drive identification of the disk containing the SELECT.COM file. If you omit the disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the SELECT.COM file. If you omit the subdirectory name, DOS uses the current default directory.

A: | B: specifies the source drive of the keyboard files.

d: is the target drive to which international files are copied.

p is the target path for the file copy.

country is one of the following 3-digit codes that specifies the country to use.

001	United States	002	French-Canadian
003	Latin America	031	Netherlands
032	Belgium	033	France
034	Spain	039	Italy
041	Switzerland	044	United Kingdom
045	Denmark	046	Sweden
047	Norway	049	Germany
061	English	351	Portugal
358	Finland	785	Arabic countries
972	Israel		

keyboard is the 2-character identifier of the keyboard layout to use (see KEYB).

NOTES SELECT uses the DISKCOPY command (DOS version 3.2 uses XCOPY) to make acopy of the DOS disk. SELECT creates the CONFIG.SYS and AUTOEXEC.BAT files that are required for international support on the target disk.

Upon completion, SELECT places the line

```
COUNTRY=xxx
```

in CONFIG.SYS and

```
KEYB XX
```

in AUTOEXEC.BAT on the target disk.

SELECT supports single- or dual-floppy drive systems.

EXAMPLES The following command directs SELECT to create a disk that fully supports German characteristics:

```
C> SELECT A: 049 GR
```

MESSAGES If SELECT displays the message

```
Cannot execute FILENAME.EXT
```

it cannot execute FORMAT, DISKCOPY, or XCOPY, which SELECT uses to create your bootable international disk. If this error occurs, make sure that the file containing the specified command is available in the current directory or command path.

If SELECT displays the message

```
File creation error
```

it could not create a file on the target disk. In most cases, this error is due to an existing subdirectory with the same name as the file on the disk or a full root directory.

If SELECT displays the message

```
Incorrect DOS version
```

the executable SELECT.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct SELECT.COM for the executing version of DOS.

If SELECT displays the message

```
Incorrect number of parameters
```

your SELECT command line contains either too many or too few parameters. Review the format of the SELECT command and reissue the command.

If SELECT displays the message

```
Insert DOS diskette in drive A
Strike any key when ready
```

it is prompting you to insert the Program Disk that you use to boot DOS in drive A. SELECT uses this disk to create a bootable international disk.

If SELECT displays the message

```
Insert KEYBxx.COM diskette in drive n:
Strike any key when ready
```

it is prompting you to insert the disk containing the specified KEYBxx.COM file into the given driven. SELECT will then copy this file to the international disk that you are building. Once you place this disk in the specified drive, press any key to continue.

If SELECT displays the message

```
Insert target diskette in drive n:
Strike any key when ready
```

it is prompting you to place the disk that will contain your international configuration into the specified disk drive so that it can complete its processing. Once you have placed the disk into the specified drive, press any key to continue.

If SELECT displays the message

```
Invalid country code
```

the international country code (3-digit number) that you have specified in the SELECT command line is invalid. Review the list of valid country codes and reissue the SELECT command.

If SELECT displays the message

```
Invalid parameter
```

one of the parameters in your SELECT command line is invalid. Review the format of the SELECT command and reissue the command.

If SELECT displays the message

```
Is KEYBxx.COM on another
diskette (Y/N)?
```

it could not find the specified KEYBxx.COM file on the current disk and is prompting you to determine whether the file resides on a different disk. If you type **Y**, SELECT will prompt you to insert the disk containing the KEYBxx.COM file. If you type **N**, SELECT will terminate its processing.

If SELECT displays the message

```
Invalid keyboard code
```

the 2-letter keyboard code that you have specified in the SELECT command line is invalid. Review the list of valid keyboard codes and reissue the correct SELECT command.

If SELECT displays the message

```
Keyboard routine not found
```

it could not find the KEYBxx.COM file on the disk that you just inserted following the SELECT prompt to insert the disk containing the KEYBxx.COM file. Therefore, SELECT will terminate its processing.

If SELECT displays the message

```
SELECT is used to install DOS the first
time. SELECT erases everything on the
specified target and then installs DOS.
Do you want to continue (Y/N)?
```

it is warning you that if you proceed with the command, it will erase all of the files on the target disk, creating a bootable disk with the international support specified. If you want to continue this command, type **Y** and press ENTER; otherwise, type **N** to terminate the command.

If SELECT displays the message

```
Unable to copy keyboard routine
```

it was unable to copy the KEYBxx.COM file to the target disk. In most cases, this error occurs when the target disk is damaged.

If SELECT displays the message

```
Code page switching not supported for this country
```

the specified country code does not support codepage switching. In most cases, users will not require codepage switching support.

If SELECT displays the message

```
Read error, KEYBOARD.SYS
```

it could not read the KEYBOARD.SYS file. In most cases, this error message indicates that the KEYBOARD.SYS file has become damaged. Retry the SELECT command with the original KEYBOARD.SYS file.

If SELECT displays the message

```
Invalid signature in KEYBOARD.SYS file
```

the KEYBOARD.SYS file may have become damaged. Retry the SELECT command with the original KEYBOARD.SYS file.

If SELECT displays the message

```
Invalid signature in COUNTRY.SYS file
```

the COUNTRY.SYS file may have become damaged. Retry the SELECT command with the orginal copy of COUNTRY.SYS.

SET

FUNCTION Places or displays DOS environment entries.

COMMAND TYPE Internal

FORMAT

SET [*name*=[*value*]]

where *name* is the name of a DOS environment entry to which you are assigning a value, and *value* is the character string that defines the value assigned.

NOTES Each time DOS boots, it reserves an area of memory called the environment.

The DOS SET command sets or displays entries in the DOS environment. The DOS environment provides a storage location for system specifics. DOS commands such as PROMPT and PATH place entries in the environment. The SET command with no parameters displays the current environment.

SET converts all entry names to uppercase.

DOS version 3.3 supports named parameters that use environment entries in conjunction with DOS batch files.

EXAMPLES The following SET command, with no command-line parameters, directs SET to display the current environment entries:

```
C> SET

COMSPEC=C:\COMMAND.COM
PATH=C:\DOS
```

In this command, SET creates a new environment entry called FILE and assigns it the value TEST.DAT:

```
C> SET FILE=TEST.DAT
```

You can verify this by again issuing the SET command, as shown here:

```
C> SET
COMSPEC=C:\COMMAND.COM
PATH=C:\DOS
FILE=TEST.DAT
```

If you are using DOS version 3.3, you can create a named parameter called FILE in a DOS batch file, as shown here:

```
CLS
TYPE %FILE%
```

When DOS encounters %FILE%, it searches the environment for a matching entry (in this case TEST.DAT) and if found displays the file's contents.

To remove the value for an entry, use SET as shown here:

```
C> SET FILE=
```

Invoking SET now displays

```
C> SET
COMSPEC=C:\COMMAND.COM
PATH=C:\DOS
```

MESSAGES If SET displays the message

```
Out of environment space
```

the environment is full. Until you install memory-resident software such as PRINT, GRAPHICS, or a third-party product, DOS allows your environment to grow to up to 32K as your processing needs require. If, however, you install memory-resident software, your environment size becomes fixed. Therefore, try to invoke all of your SET commands prior to loading memory-resident software. Also, remember that the SHELL= entry in CONFIG.SYS allows you to specify the environment size.

SHARE

FUNCTION Supports DOS file sharing.

COMMAND TYPE External SHARE.EXE

FORMAT

[*drive:*][*path*]SHARE [/F:*file_space*][/L:*locks*]

where the following is true:

drive: is the disk drive identification of the disk containing the SHARE.COM file. If you omit the disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the SHARE.COM file. If you omit the directory name, DOS uses the current default directory.

/F:*file_space* allocates memory (in bytes) for the area in which DOS stores file-sharing information. Each open file requires 11 bytes, plus the length of the file name (up to 63 characters). The default file space is 2048 bytes.

/L:*locks* allocates memory for the desired number of file locks. The default value is 20.

NOTES DOS versions 3.0 and greater support file and record locking. Each time a file is opened with file sharing installed, DOS checks to see if the file is locked against the open operation. If so, the file cannot be opened. In addition, DOS checks for locking during each read or write operation.

SHARE is a memory-resident software program that performs file and record locking; therefore, it can be installed only one time.

Once installed, the only way to remove file sharing is to reboot. SHARE places considerable overhead on all of your file operations; therefore, install SHARE only when file sharing is in effect.

EXAMPLES The following command invokes file sharing with the default values of 2048 and 20 locks:

```
C> SHARE
```

This command installs file-sharing support with 40 locks:

```
C> SHARE /L:40
```

MESSAGES If SHARE displays the message

```
Incorrect DOS version
```

the executable SHARE.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct SHARE.EXE for the executing version of DOS.

If SHARE displays the message

```
Incorrect parameter
```

your command line very likely contains an invalid qualifier. Examine the SHARE format along with the valid values for /L and /N, and reissue the command.

If SHARE displays the message

```
Not enough memory
```

your computer does not have enough memory to invoke the SHARE memory-resident software. If possible, remove other memory-resident software and RAM drives.

SHIFT

FUNCTION Shifts each batch parameter one position to the left.

COMMAND TYPE Internal

FORMAT

SHIFT

NOTES If more than ten parameters are passed to a DOS batch procedure, you can use the SHIFT command to access each parameter past %9. If no parameter exists to the right of a parameter, SHIFT assigns the parameter a NULL string.

EXAMPLE The following batch file displays all of the batch param-eters specified on the command line:

```
ECHO OFF
:LOOP
SHIFT
IF '%0'== '' GOTO DONE
ECHO %0
GOTO LOOP
:DONE
```

If the previous file was named TEST.BAT, invoking the batch file as

```
C> TEST 1 2 3 4
```

would display

```
C> ECHO OFF
1
2
3
4
```

SORT

FUNCTION Activates the DOS SORT filter.

COMMAND TYPE External SORT.EXE

FORMAT

DOS_command | [*drive:*][*path*]SORT [/R][/+*n*]

or

[*drive:*][*path*]SORT [/R][/+*n*] < *file*

where the following is true:

drive: is the disk drive identification of the disk containing the SORT.EXE file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the SORT.EXE file. If you do not specify a directory name, DOS uses the current default directory.

/R directs SORT to sort the data in reverse order.

/+*n* allows you to specify the column on which to sort the data.

NOTES The DOS SORT command reads data from the standard input device, sorting and displaying the information to the standard output device until the end of file.

You can use SORT as a filter with the DOS redirection operators < or |.

EXAMPLES The following command directs SORT to sort the information contained in the DATA.DAT file:

```
C> SORT < DATA.DAT
```

This command directs SORT to sort the same file, this time in reverse order:

```
C> SORT /R < DATA.DAT
```

Assume that your data file contains the following:

Bill M
Mary M
Kris M
Kal M
Andy F
Jim M
Mike M
Ed M

The following command directs SORT to sort the file based upon the data starting in column 5:

```
C> SORT /+6 < FILENAME.EXT
```

SORT displays

```
Mary  F
Andy  F
Bill  M
Kris  M
Kal   M
Jim   M
Mike  M
Ed    M
```

If you do not redirect input to SORT, as shown here

```
C> SORT
```

SORT expects its input from the keyboard. Type the following letters and then press CTRL-Z:

```
C> SORT
A
F
D
E
B
C
^Z
```

SORT sorts the data and displays

```
C> SORT
A
F
D
E
B
C
^Z
A
B
C
D
E
F

C>
```

MESSAGES If SORT displays the message

```
Invalid parameter
```

one of the qualifiers in your SORT command is invalid. SORT supports only /R and /+n.

If SORT displays the message

```
Incorrect DOS version
```

the executable SORT.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct SORT.COM for the executing version of DOS.

If SORT displays the message

```
SORT: Insufficient disk space
```

the current disk does not contain adequate disk space for DOS to create the temporary files required for DOS I/O redirection, or a redirected file is too large to fit on disk.

If SORT displays the message

Insufficient memory

your system does not have enough RAM for the SORT command to execute. If you have installed memory-resident software or a RAM drive, consider restarting DOS without them, so that you can reissue the SORT command.

SUBST

FUNCTION Substitutes a drive name for a DOS path name.

COMMAND TYPE External SUBST.EXE

FORMAT

[*drive:*][*path*]SUBST [*d:*] [*pathname*][/D]

where the following is true:

drive: is the disk drive identification of the disk containing the SUBST.EXE file. If you do not specify a disk drive, DOS uses the current default disk drive.

path is the name of the DOS directory that contains the SUBST.EXE file. If you do not specify a directory name, DOS uses the current default directory.

d: is the disk drive identification that will be used to reference the path.

pathname is the DOS path name to abbreviate.

/D directs SUBST to remove a previous disk substitution.

NOTES Because DOS path names can become quite large, DOS allows you to substitute a drive identifier for a path name.

If you invoke SUBST without any parameters, current substitutions are displayed.

If you plan to perform multiple disk drive substitutions, you must use the LASTDRIVE= entry in CONFIG.SYS.

EXAMPLES After you issue the following command, DOS will allow you to abbreviate the \DOS\HELPFILE\COMMANDS subdirectory to the drive letter E:

```
C> SUBST E:\DOS\HELPFILE\COMMANDS
```

The following command displays the contents of the directory called \DOS\HELPFILE\COMMANDS:

```
C> DIR E:
```

If that directory contains other directories, you can still use the drive letter as shown here:

```
C> DIR E:SUBDIR
```

If you invoke SUBST without any command-line parameters, SUBST displays the current substitutions, as shown here:

```
C> SUBST
E: => C:\DOS\HELPFILE\COMMANDS
```

MESSAGES If SUBST displays the message

```
Cannot SUBST a Network drive
```

the SUBST command failed because the disk drive specified in the command line refers to a network disk. DOS does not allow you to substitute a network disk drive.

If SUBST displays the message

```
Incorrect DOS version
```

the executable SUBST.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct SUBST.EXE for the executing version of DOS.

If SUBST displays the message

```
Incorrect number of parameters
```

your SUBST command line contains either too many or too few command-line parameters. Make sure that you have specified a disk drive and directory, or a disk drive and the /D qualifier.

If SUBST displays the message

```
Invalid parameter
```

make sure that your command line contains a valid qualifier and that your disk drive letter is valid. If you are using a disk drive letter greater than E, you must use the LASTDRIVE= entry in CONFIG.SYS to specify additional disk drive letters.

If SUBST displays the message

```
Path not found
```

SUBST could not locate the subdirectory specified in your command line to substitute.

If SUBST displays the message

```
Not enough memory
```

your system does not have enough memory to execute the command. If you have installed memory-resident software or a RAM drive, reboot DOS without them and reissue the SUBST command.

SYS

FUNCTION Transfers to the target disk the hidden operating system files that perform the initial system startup processing.

COMMAND TYPE External SYS.COM

FORMAT

[*drive:*][*path*]SYS *target_drive:*

where the following is true:

drive: specifies the disk drive containing the SYS.COM file. If you do not specify a disk drive id, DOS uses the current default disk drive.

path is the DOS path name of the directory that contains the SYS.COM file. If you do not specify a DOS path name, DOS uses the current default directory.

target_drive: specifies the target disk drive for the hidden operating system files.

NOTES SYS does not copy the COMMAND.COM file to the target disk. You must do this with the DOS COPY command.

SYS does not work when a JOIN or SUBST command is in effect.

DOS transfers files only to an empty target disk or to a disk previously formatted with the /S or /B qualifier.

EXAMPLE The following SYS command transfers the hidden operating system files to the disk in drive A:

```
C> SYS A:
```

Verify the procedure by issuing the CHKDSK command, as shown here:

```
C> CHKDSK A:

362496 bytes total disk space
53248 bytes in 2 hidden files
25600 bytes in 1 user files
283648 bytes available on disk

655360 bytes total memory
586912 bytes free
```

MESSAGES If SYS displays the message

```
Cannot SYS to a Network drive
```

the command failed because DOS will not allow you to copy system files to a network device.

If SYS displays the message

```
Incompatible system size
```

it could not copy the hidden system files as contiguous files on the target disk. The disk may not be bootable.

If SYS displays the message

```
Incorrect DOS version
```

the executable SYS.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct SYS.EXE for the executing version of DOS.

If SYS displays the message

```
Insert destination disk in drive n
and strike any key when ready
```

it is prompting you to insert the disk that you want SYS to transfer the system files to into the specified disk drive.

If SYS displays the message

```
Insert system disk in drive n
and strike any key when ready
```

it is prompting you to insert a Program Disk in the specified disk drive so that it can copy the hidden system files from the disk to the target disk specified in your SYS command line.

If SYS displays the message

```
Invalid drive specification
```

the target disk drive in your SYS command line is invalid. Make sure that you have placed a colon after the drive letter and that you are using a valid disk drive (A, B, C, and so on).

If SYS displays the message

```
Invalid parameter
```

you have very likely not included a disk drive in your SYS command line.

If SYS displays the message

```
No room for system on destination disk
```

it could not place the hidden system files on the specified target drive. The hidden DOS files must be the first files that DOS places on a disk. If SYS displays this message, you will very likely have to use FORMAT /S to create a bootable system disk. If you are using a fixed disk, make sure that you completely back up all of the files on your disk before doing so. Once the FORMAT /S command completes, you can restore the original files back to the disk drive, adding the required DOS command files.

If SYS displays the message

```
No system on default drive
Insert system disk in drive n
and press any key when ready
```

it could not find the hidden files on the disk in the current drive. Place a bootable Program Disk in the specified drive and press ENTER.

TIME

FUNCTION Sets the DOS system time.

COMMAND TYPE Internal

FORMAT

TIME [*HH:MM* [:*SS* [.*hh*]]]

where the following is true:

HH is the desired hours (0-23).
MM is the desired minutes (1-59).
SS is the desired seconds (0-59).
hh is the desired hundredths of seconds (0-99).

NOTES If you do not specify a time, TIME displays the current time.

In the past, most DOS users had to set their system clocks by using the Setup disk provided in the *Guide to Operations* manual. The DOS version 3.3 TIME command modifies the AT system clock, thus removing this requirement.

EXAMPLES This command does not specify a time on the command line:

```
C> TIME
```

TIME prompts you for one, as shown here:

```
Current time is 16:08:41.15
Enter new time:
```

To leave the time unchanged, press ENTER; otherwise, type the desired time.

This command sets the clock to 12 noon:

```
C> TIME 12:00
```

This command sets the clock to midnight:

```
C> TIME 00:00:00.00
```

If the time that you specify is invalid, for example

```
C> TIME 15:65:00
```

TIME displays an error message and reprompts you for the time, as shown here:

```
Invalid time
Enter new time:
```

MESSAGES If TIME displays the message

```
Invalid parameter
```

one of the separators that you are using between the hours, minutes, and seconds is invalid. TIME supports only the colon and period.

If TIME displays the message

```
Invalid time
Enter new time:
```

the time that you have typed is invalid. Enter a new time in the format hh:mm:ss.nn and press Enter. TIME only requires you to enter the hours and minutes.

TREE

FUNCTION Displays a directory structure.

COMMAND TYPE External TREE.COM

FORMAT

[*drive:*][*path*]TREE [*d:*][/F]

where the following is true:

drive: specifies the disk drive identification of the disk containing the TREE.COM file. If you do not specify a disk drive, DOS uses the current default disk drive.

path specifies the name of the DOS directory that contains the TREE.COM file. If you do not specify a subdirectory, DOS uses the current default directory.

d: is the disk drive identification of the disk directory TREE is to display.

/F directs TREE to also display the name of each file in a directory.

NOTES By default, TREE displays the name of each directory on a disk. To also display each file name, use the /F qualifier.

EXAMPLES The following command displays the directory structure of the disk:

```
C> TREE
```

The output is similar to

```
Directory path listing

Path: \IBM

Subdirectories:  SALES


Path: \IBM\SALES

Subdirectories:  None


Path: \MS

Subdirectories:  None
```

Use the following command to also display the files in each directory:

```
C> TREE B:/F
```

When you invoke the command, your display is similar to this:

```
Directory path listing

Files:        None


Path: \IBM

Subdirectories:  SALES

Files:        NOTES  .TXT


Path: \IBM\SALES

Subdirectories:  None

Files:        JULY   .DAT
```

```
Path: \MS

Subdirectories: None

Files:      C     .TXT
PRES   .TXT
```

MESSAGES If TREE displays the message

```
Incorrect DOS version
```

the executable TREE.COM file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct TREE.COM for the executing version of DOS.

If TREE displays the message

```
Invalid drive specified
```

the disk drive letter that you have placed in the TREE command line is invalid. Make sure that you have placed a colon after the disk drive name and that you are using an existing disk drive (A, B, C, and so on).

If TREE displays the message

```
Invalid parameter
```

your command line contained an invalid qualifier, or also included a file or path name in addition to the disk drive letter. Refer to the TREE format and reissue the command.

If TREE displays the message

```
No subdirectories exist
```

it is telling you that it did not find any DOS subdirectories on the specified disk. This message is purely informational.

If TREE displays the message

```
Invalid path
```

it could not process the specified subdirectory. Use the CHKDSK /F command to repair the disk and then repeat the TREE command.

TYPE

FUNCTION Displays a text file's contents.

COMMAND TYPE Internal

FORMAT

```
TYPE file_specification
```

where *file_specification* is the complete DOS file specification for the file to be displayed. It can contain a disk drive identification and a DOS path name.

NOTES TYPE is restricted to ASCII files. Do not use TYPE on COM or EXE files because these files contain unprintable characters that cause your screen to beep and display uncommon characters.

EXAMPLES The following command directs TYPE to display the contents of the CONFIG.SYS file:

```
C> TYPE CONFIG.SYS
```

The command

```
C> TYPE B:\DOS\AUTOEXEC.SAV
```

directs TYPE to display the contents of the AUTOEXEC.SAV file in the DOS directory on drive B.

If the file specified in the TYPE command line does not exist, TYPE displays

```
Invalid filename or file not found
```

MESSAGES If TYPE displays the message

```
File not found
```

the file name that you have specified in the TYPE command line does not exist. Check your spelling and the current disk drive and subdirectory.

If TYPE displays the message

```
Invalid drive specified
```

the disk drive letter that you have placed before the file to display is invalid. Make sure that you have placed a colon after the disk drive letter and that you are using an existing drive letter (A, B, C, and so on).

If TYPE displays the message

```
Invalid path or file name
```

either the subdirectory name or file specified in the TYPE command line is invalid. Make sure that the subdirectory path name exists on the disk as specified and reissue the command.

VER

FUNCTION Displays the DOS version number.

COMMAND TYPE Internal

FORMAT

```
VER
```

NOTES DOS version numbers are composed of one major and one minor version number. For example, DOS version 3.2 has a major version number of 3 and a minor version number of 2.

EXAMPLE The following command directs VER to display the current version number:

```
C> VER
```

For DOS version 3.3, the output is

```
MS-DOS Version 3.30
```

VERIFY

FUNCTION Enables or disables disk verification.

COMMAND TYPE Internal

FORMAT

VERIFY [ON | OFF]

where ON enables DOS disk verification, and OFF disables DOS disk verification.

NOTES Periodically, a disk drive may not correctly record the information on disk as DOS intended. Although very rare, such occurrences can leave incorrect data on disk.

If you enable disk I/O verification, DOS will double-check the data it writes to disk by rereading each sector and comparing it to the original data. If a discrepancy exists, DOS can detect it.

If you invoke VERIFY without a command-line parameter, VERIFY displays its current state, ON or OFF.

Because VER requires DOS to reread each sector that it writes to disk, disk verification has significant system overhead.

EXAMPLES The following command enables disk I/O verification:

```
C> VERIFY ON
```

If you invoke VERIFY without command-line parameters, VERIFY displays its current state, ON or OFF, as shown here:

```
C> VERIFY
VERIFY is on.

C>
```

MESSAGES If VERIFY displays the message

```
Must specify ON or OFF
```

you have invoked the VERIFY command with a command-line parameter other than ON or OFF. Reissue the command with no command-line parameters (to

display the current verify setting), or with either the word ON (to enable disk verification) or OFF (to disable disk verification).

VOL

FUNCTION Displays a disk volume label.

COMMAND TYPE Internal

FORMAT

VOL [*drive:*]

where *drive:* specifies the disk drive identification of the disk volume VOL is to display. If you do not specify a disk identification, VOL will use the current default disk drive.

NOTES DOS volume labels are 11-character names assigned to a disk.
Volume names use the same characters as DOS file names.
If you do not specify a drive, VOL uses the current default.
To assign a volume label, use the DOS LABEL command.

EXAMPLE In the following command, VOL displays the disk volume label of the disk contained in the current drive:

```
C> VOL
Volume in drive C is DOSDISK
```

MESSAGES If VOL displays the message

```
Invalid drive specified
```

the disk drive that you have specified in the VOL command line is invalid. Make sure that you have placed a colon after the device name and that you are using an existing disk drive (A, B, C, and so on). If you simply invoke VOL without specifying a disk drive, VOL will display the volume label of the current disk drive.

If VOL displays the message

```
Volume in drive n has no label
```

it is telling you that DOS has never assigned a label to the disk in the specified drive. To assign a label to a disk, use the DOS LABEL command.

XCOPY

FUNCTION Copies source files and directories to a target destination.

COMMAND TYPE External XCOPY.EXE

FORMAT

[*drive:*][*path*]XCOPY *source_filespec* [*target_filespec*]
[/A][/D:*mm-dd-yy*][/E][/M][/P][/V]
[/S][/W]

where the following is true:

drive: specifies the disk drive containing the XCOPY.EXE file. If you do not specify a disk drive id, DOS uses the current default disk drive.

path is the path name of the directory that contains the XCOPY.EXE file. If you do not specify a path name, DOS uses the current default directory.

source_filespec is the complete file specification for the source files that XCOPY is to copy.

target_filespec is the destination name for the files copied by XCOPY.

/A directs XCOPY to copy only files that have their archive bit set.

/D:*mm-dd-yy* directs XCOPY to copy only files created since the specified date.

/E directs XCOPY to place directories on the target disk if the directory is currently empty.

/M functions like the /A qualifier; however, /M directs XCOPY to clear each file's archive bit as it copies them.

/P directs XCOPY to prompt

```
FILENAME.EXT (Y/N)?
```

before copying each file.

/V directs XCOPY to compare the contents of the target file and the source file to ensure that the file copy was successful.

/S directs XCOPY to copy the contents of lower-level directories to the target location.

/W directs XCOPY to prompt

```
Press any key to begin copying file(s)
```

before beginning.

NOTES XCOPY is more useful than COPY and DISKCOPY because it selectively copies files contained in DOS directories. In fact, many users use XCOPY to repair disk fragmentation or as a system backup mechanism.

EXAMPLES If you have a blank disk in drive B, the following command creates a disk structure on the disk in drive B identical to that of the disk in drive A:

```
C> XCOPY A:\*.* B:\ /S
```

The following command copies all of the files contained in or below the directory \A on drive A to the disk in drive B:

```
C> XCOPY A:\A\*.* B:\ /S
```

To copy the entire contents of a fixed disk to floppy disks, first set the attribute of each file on the fixed disk to indicate that each requires a backup, as shown here:

```
C> ATTRIB +A \*.* /S
```

Next, issue the command

```
C> XCOPY \*.* A:\ /M /E /S
```

XCOPY begins transferring files to the floppy disk while maintaining the existing disk structure. When the target disk becomes full, insert a new floppy disk in drive A and invoke this command:

```
C> XCOPY \*.* A:\ /M /E /S
```

XCOPY continues where it left off, because it has been clearing the archive bit on each file it successfully copies to the target disk.

MESSAGES If XCOPY displays the message

```
FILENAME.EXT File not found
```

it could not locate the source file specified in your XCOPY command line. Recheck your spelling and the current directory.

If XCOPY displays the message

```
drive:path\filename.ext (Y/N)?
```

you have invoked the XCOPY command with the /P qualifier, which allows you to perform selective file-copy operations. If you do not want to copy this file, respond with **N** to copy the file. Type **Y** and press ENTER.

If XCOPY displays the message

```
Access denied
```

either the source file specified in your XCOPY command is locked by another application or the target file name has been marked as read-only.

If XCOPY displays the message

```
Cannot COPY from a reserved device
```

your XCOPY command line specifies a device, such as COM1, as the source of the file copy. XCOPY does not support this type of operation. Use the DOS COPY command.

If XCOPY displays the message

```
Cannot COPY to a reserved device
```

your XCOPY command line specifies a device, such as COM1, as the target of the file copy. XCOPY does not support this type of operation. Use the DOS COPY command.

If XCOPY displays the message

```
Cannot perform cyclic copy
```

your XCOPY command line contains the /S qualifier and your source and target specifications are such that a subdirectory specified in the target is also a subdirectory in the target. XCOPY will read from the subdirectory, writing files back to the directory, which causes the operation to continue in a circular fashion. XCOPY does not support this type of operation.

If XCOPY displays the message

```
Does FILENAME specify a file name
or directory name on the target
(F=file, D=directory)?
```

it could not determine whether the target of the file-copy operation is supposed to be a file name or a directory. If you want XCOPY to write all of the information to a file, type **F** and press ENTER. If you want XCOPY to create a subdirectory and place the files into it, type **D** and press ENTER.

If XCOPY displays the message

```
File cannot be copied onto itself
```

the source and target locations of the XCOPY command are the same. XCOPY does not support this type of operation. You must either change the target disk drive id or the subdirectory.

If XCOPY displays the message

```
File creation error
```

it could not create the specified target file because either the root directory of the target disk is full or a subdirectory with the same name as the file already exists on disk.

If XCOPY displays the message

```
Incorrect DOS version
```

the executable XCOPY.EXE file that you invoked is from a different version of DOS than is currently executing. Either invoke the corresponding version of DOS or locate the correct XCOPY.EXE for the executing version of DOS.

If XCOPY displays the message

```
Insufficient disk space
the target disk is full.
```

several files may have not yet been copied.

If XCOPY displays the message

```
Invalid date
```

the date that you have specified in conjunction with the /D qualifier is invalid. Make sure that you have specified your date in the format mm-dd-yy and reissue the command.

If XCOPY displays the message

```
Invalid drive specification
```

one of the disk drives in your XCOPY command line is invalid. Make sure that you have placed a colon following the disk drive letter and that the letter you are specifying is a valid disk drive (A, B, C, and so on).

If XCOPY displays the message

```
Invalid number of parameters
```

your XCOPY command contains either too few or too many qualifiers. Check the format of the XCOPY command and reissue the command.

If XCOPY displays the message

```
Invalid parameter
```

make sure that all of the qualifiers in your command line are valid and reissue the XCOPY command.

If XCOPY displays the message

```
Invalid path
```

either your source or target path name is invalid. Recheck your command line and reissue the command.

If XCOPY displays the message

```
Insufficient memory
```

your system does not have enough memory to execute the command. If you have installed memory-resident software or a RAM drive, reboot DOS without them and reissue the XCOPY command.

If XCOPY displays the message

```
Path too long
```

either your source or target path name exceeds 63 characters. DOS only supports path names of up to 63 characters.

If XCOPY displays the message

```
Press any key to begin copying file(s)
```

your XCOPY command line contains the /W qualifier, which suspends processing until you press ENTER to start the command. This allows you to change disks as required.

If XCOPY displays the message

```
Reading source file(s)
```

it is telling you that it is reading the files from the source disk to determine which files to copy to the target location.

If XCOPY displays the message

```
Unable to create directory
```

it is unable to create the specified subdirectory on the target disk because either the root directory of that disk is full or a file already exists on that disk with the same name.

Trademarks

AT™	International Business Machines
Commodore®	Commodore Business Machines, Inc.
IBM®	International Business Machines
Intel®	Intel Corporation
Lotus®	Lotus Development Corporation
Microsoft®	Microsoft Corporation
MS-DOS®	Microsoft Corporation
PS/2™	International Business Machines
WordStar®	MicroPro International
XT™	International Business Machines

Index

\, to represent root directory, 342
% sign, for batch-file parameters, 305
%0, as batch-file name, 305
* (asterisk) in file names. *See* Wildcard characters
. and .. directories, 358-360
: (colon), in batch-file label names, 316
? (question mark) in file names. *See* Wildcard characters
@ sign, to suppress command-name display, 295-296

A

Adapters, multiple-plug, 6
Aligning text on the screen, 137-138
ALT, enhancing files with, 172-177
ANSI cursor-control commands, 266
ANSI escape sequences, 261-267
    changing screen color with, 263-265
    displaying messages with, 265-266
    for reassigning keys, 266-267
    to clear the screen, 262-263
    used with PROMPT, 466
    values for setting screen attributes, 264
ANSI.SYS device driver, 253
    having fun with, 260-267
ANSI.SYS driver, *continued*
    installing, 260-261
    setting screen color, 303
APPEND command, 384-387, 549-553
    /E qualifier, 387
    /X qualifier, 386
Append redirection operator (>>), 187-189
Appending redirection, 187, 189
    and creating target file, 188
Appending text to a file, with COPY +, 441-443
Application programs
    returning to from DOS, 471
    updating, 430-435
Archive required attribute, 419
    setting, 450-452
Archive required bit, 419
ASCII character set, 174-176, 545-548
ASSIGN command, 398-399, 554-556
ATTRIB command, 409, 416-422, 556-560
    +A qualifier, 419
    -A qualifier, 419
    +R qualifier, 417-418, 420-421
    -R qualifier, 418
    /S qualifier, 421-422
    with XCOPY command, 450
Attributes, file. *See* File attributes

AUTOEXEC.BAT file
a special batch file, 334
PATH command in, 382
PRINT command in, 229
PROMPT command in, 466-467
AUX port, 132, 467-468

B

Backing up disks and files, 503-526
(*See also* Copying files)
and inserting new disks, 510
fixed disk to floppies, 506
1.2MB disk, 510
360K disk, 510
using formatted disks for, 508-509
Backslash (\), 342
BACKSPACE
deleting with, 150-152
location of, 151
BACKUP command, 503-527, 560-566
/A qualifier, 512, 519
exit status values, 522
/F qualifier, 508
format of, 507
/L qualifier, 520-521
parameters to, 507-508
when to use, 505-506
Backup copies
daily, 518-520
developing a policy, 516-520
making, 15
monthly, 518-519
BACKUP, FORMAT, RESTORE operation, 524-526
Bad sectors, 90
BAT extension, 287
Batch-file names, assigned by DOS, 305
Batch files
as text files, 287
BACKUP and RESTORE in, 521-522
building intelligent, 311-334
debugging with DEBUG, 531-543
documenting, 298-300
executing, 288, 332-333
for the DATE and TIME commands, 288-289, 296-297
invalid commands in, 289-290
naming, 290-291
nesting, 331-333
order of commands in, 289
order of execution, 311
suppressing command-name display in, 294-295
suspending, 300-301, 471-472
terminating, 291, 293-294
to display messages, 296-297
to reassign keys, 306-307
to set screen attributes, 297-298, 301-302, 306
using, 287-307
Batch parameters, 301-307
named, 328-331
testing for nonexistent, 320-321
using multiple, 324-328
Baud rate, 133
setting with MODE command, 140
Bell, program to sound computer's, 532-536
Block device attributes, setting, 243

Block mode devices, 243
Bootable disk, creating, 92-94
Booting DOS, 11
Brackets, in command lines, 51
BREAK command, 237-238, 567-568
    BREAK OFF command, 237
    BREAK ON command, 236-237
BREAK= entry in CONFIG.SYS, 234, 238
BUFFERS= entry in CONFIG.SYS, 234, 241-242
Bugs
    explanation of, 153
    in batch files, 531
Byte, definition, 59

C

CALL command, 569-570
    for executing batch files, 332-333
Canceling a command, 152-153
Case, of DOS commands, 26
Case matching, in IF string conditions, 314
CD command. *See* CHDIR command
Character echoing, 164
Character mode devices, 243
Character sets, foreign, 486
CHCP command, 494-495, 497, 570-572
CHDIR command, 346-350, 572-575
    to display the current directory, 349-350
CHKDSK command, 106, 111-114, 516-518, 575-584
    compared with DIR, 108
    /F qualifier, 108
    /V qualifier, 107, 375-376
Clearing the screen, 25-26
    with ANSI escape sequences, 262-263
Clock ticks, specifying, 228
CLS command, 25-26, 42, 584
Codepage information (CPI) files, 496
Codepage switching, 492-495
Codepages
    as character sets, 486
    available to CHCP command, 494-495
    displaying the current, 476, 495, 498
    for printing, 496-497
    for screen display, 496
    foreign, 486
COM and EXE files, displaying, 124
COM1 port, 130-132, 161-162
    as output to the PRINT command, 212-213
Combining files, 440-441
Command buffer, 166
    editing text in, 168-172
COMMAND command, 585-587
    (*See also* COMMAND.COM)
    /C qualifier, 332-333
Command files
    defining a path to external, 277-278
    locating, 376-382
    on virtual disks, 260
Command line
    copying characters to, 166, 168, 170
    copying commands to, 167-168
    definition, 50
    deleting characters on, 170-172

Command line, *continued*
editing the current, 172
inserting characters on, 169-170
Command-line arguments, 50-51
Command-line interpreter, 250-251, 277
Command-line parameters
introduction to, 50-51
optional, 51
COMMAND.COM, 29, 121
alternatives to, 250-251
and bootable disks, 93-94
copying to the root directory, 383
Command output, redirecting with >, 183-186
Command reference, 549-746
Command search path, removing, 381
Commands
canceling, 152-153
copying to the command line, 167-168
entering, 37-54
process of DOS executing, 292
Comments in batch files, 298-300
COMP command, 409, 422-425, 587-592
Comparing files
with the COMP command, 422-425
with the FC command, 425-430
COMSPEC= entry in DOS environment, 277-278
CON device, 121
for output, 127-128
referencing the keyboard, 127
CON device, *continued*
referencing the screen display, 128
CONFIG.SYS
BREAK= entry, 234, 238
BUFFERS= entry, 234, 241-242
COMSPEC= entry, 277-278
creating, 233-235, 383
DEVICE= entry, 234, 253-267, 492-493
DRIVPARM= entry, 234, 243-245
FCBS= entry, 234, 246
FILES= entry, 235, 248
LASTDRIVE= entry, 235, 248-249, 404-405
list of entries in, 234-235
making backup of, 235
modifying, 235
PATH= entry, 277-278, 376-377
rebooting to affect changes, 234
SHELL= entry, 235, 250-251, 277, 282
STACKS= entry, 235, 251-252
Contiguous files, 109-114
COPY command, 70-75, 439-445, 593-596
/A qualifier, 443
/B qualifier, 443
format of, 440
overwriting existing files, 126
overwriting files of the same name, 71
using to print files, 132
/V qualifier, 443

COPY command, *continued*
versus XCOPY command, 454-455
with subdirectories, 363
COPY CON command, 121-124
canceling, 125-126
to create CONFIG.SYS, 235
Copying characters to command line, 166-167
Copying commands to command line, 166-168
Copying disks, 26-31
from drive A to drive B, 28
single-disk drive system, 30
Copying files, 70-75, 439-455 (*See also* Backing up disks and files)
and verifying, 443
and verifying sectors, 444
copying multiple files, 72-74
from fixed to floppy disk, 450-452
in subdirectories, 363
omitting the target name, 74
to a disk other than the default, 70
Copyright display, DOS, 24-25, 37, 39
Country codes, 490-491
Country-specific information, defining, 243
COUNTRY= entry, 234, 243, 490-491
CTRL-ALT-DEL, 30-31, 149
CTRL-BREAK
enabling extended checking for, 236-238
location of keys, 154
CTRL-BREAK, *continued*
program to get state of, 537-538
to terminate a batch file, 291, 293-294
to terminate a program, 153, 156
CTRL-C, 125-126 (*See also* CTRL-BREAK)
to terminate a command, 155
to terminate a program, 154-156
CTRL-PRTSC
echoing to the printer, 164-165
location of keys, 165
CTRL-S
location of keys, 157
suspending output with, 156-159
with DIR command, 156-158
with TYPE command, 160
CTRL-Z, checking for, 424
CTTY command, 467, 596-597
CTTY CON command, 468
Current directory
and DIR command, 348
creating directories in, 355
displayed in system prompt, 465
explained, 345
for each disk drive, 350
removing, 353
root directory as, 347
selecting, 346-350
selecting a different, 349
. directory as, 359
Current drive, 46-48, 345

Current drive, *continued*
changing, 47-48
program to obtain, 536-537
Cursor-control commands, ANSI, 266
Customizing DOS, 233-274
Cylinders, disk, 88-89

D

Damaged files
locating, 108
recovering, 106, 412-416
Data blocks, 243
Data communication
parallel, 128-133
serial, 128-133
speed of, 133
Data-file search path
commands recognizing, 386
creating, 384-387
Database records, read into memory, 240
Databits, setting, 140
Date
changing the current, 49
displaying the current, 48
entering the current, 20-21
in directory listings, 59
of file creation, 56
of file modification, 56
setting the system, 48
suppressing in directory display, 60, 62
using the correct, 24
Date and time stamps, 24
DATE command, 48-49, 597-599
with parameters, 51
Date prompt, 20
Day of the week, determining for a date, 49-50
Dead keys, 489
DEBUG
to create programs, 532-543
to debug batch files, 531-543
DEBUG prompt, 533
Default drive. *See* Current drive
DEL command, 78-80, 350, 591-601
and read-only files, 79
and write-protected disks, 79
DEL key, 170-171
Deleting characters
with BACKSPACE key, 150-152
with DEL key, 170-171
Deleting files, 78-80
Deleting multiple files, 79-80
Device attributes, setting, 243
Device drivers, installing, 253-267
Device names, 128-132
DEVICE= entry in CONFIG.SYS, 234, 253-267, 491-493
Devices, configuring, 133-134
DIR command, 56, 601-605
and complex directory structures, 372
and the current directory, 348
error messages with, 65
output to a file, 186
output to a printer, 185
/P qualifier, 60-62, 156-157
using default I/O, 185
/W qualifier, 60-62
with file name, 62-63
with subdirectories, 362
<DIR> display, 343
Directories (*See also* Subdirectories)
as lists of files, 56
number of files in, 75

Directories, *continued*
of other than default drives, 63-64
rules for naming, 344-345, 358
. directory 358-360
.. directory, 358-360
Directory listings
first two lines of, 64
five-across, 60, 62
of files on disk, 57
of subdirectory files, 361-362
resumed, 62
suspending scrolling of, 60-61
Directory names, rules for, 344-345, 358
Directory strings, 357
Directory structure
complex, 372
displaying with TREE, 373-375
hardcopy of, 374, 376
Disk access speed, 109-114
Disk buffers
data stored in, 239
increasing the number of, 240-242
record found in, 241
Disk cylinders, formatting, 88-89
Disk-drive specifiers, 63-64
as targets for copy operations, 73-74
Disk drives
current directory for each, 350
default, 46-48
defining the number of logical, 248-249
exchanging files between, 94-95
installing virtual, 257-260
logical supported, 404-405
Disk drives, *continued*
making one appear as another, 397-399
1.2MB, 94
opening, 20
specifying another, 47
supporting external floppy-disk, 254-256
systems with internal floppy-disk, 254
360K and 1.2MB, 94-95
types of, 18
using logical, 393-405
Disk envelope, 13-14
Disk formatting
and losing information, 91
and making bootable, 92-94
for 1.2MB drive, 95
Disk fragmentation, 522-524
Disk-I/O
improving performance of, 468-470
operations, 238-242
Disk labels (paper), 29
writing on, 17
Disk rotation, 111-113
Disk sectors, 239-240, 413
consecutive, 110
damaged, 90, 105-107
lost, 414
Disk space
and system files, 93-94
determining, 516-518
Disk spindle, 19
Disk storage box, 16, 504
Disk storage media, 14
Disk verification, 539-540
Disk volume labels, 97-98

Disk volume labels, *continued*
deleting, 99
displaying, 99-100
in hidden files, 107
DISKCOMP command, 605-610
exit status values, 475
format of, 473
to compare two floppies, 473-475
DISKCOPY command, 26-31, 44, 611-617
on single-floppy disk system, 397
Diskettes. *See* Floppy disks
Disks (*See also* Fixed disks; Floppy disks)
appearing larger than they are, 399-402
assigning names to, 95-99
backing up, 503-526
bytes available on, 518
combining in one logical, 399-402
damaged, 28-29
examining the state of, 105-109
formatting, 86, 88-95
handling, 86-87
labeling, 515-516
maximum number of files on, 339-340
naming, 95-100
number of subdirectories on, 341
recovering damaged, 412-416
space available on, 90
DISPLAY.SYS device driver, 253, 492-493
Distribution box, hardware, 9
DOS commands
case of, 26
internal and external, 40-41
DOS environment
assigning a variable to, 279
COMSPEC= entry in, 277
displaying the default, 275-276
growth of, 280-281
increasing the size of, 251
named parameters in, 328-331
PATH= entry in, 277
placing new entries in, 278-283
removing a variable from, 279
using, 275-283
DOS Program Disk. *See* Program Disk
DOS prompt, 25, 37-38
changing, 279-280
containing default disk drive, 47
customizing, 462-467
DOS system services, 543
DOS versions prior to 2.0, 245-246
DRIVER.SYS device driver, 254
format of, 255
parameters to, 255-256
Drives, disk. *See* Disk drives
DRIVPARM= entry in CONFIG.SYS, 234
format of, 243-244
parameters to, 244-245

E

ECHO command, 617-620
in batch files, 294-295

ECHO command, *continued*
to display batch-file parameters, 302, 304
to display messages, 296-297
to set screen attributes, 297-298, 301-302
to set screen color, 306
to suppress display of remarks, 299-300
to suppress PAUSE messages, 300-301
with @ sign, 296
Echoing text to printer, 164-165
Editing the command buffer, 168-172
EDLIN
creating files of escape sequences with, 261
using to create CONFIG.SYS, 235
80286/80386 processor speed, 147
Empty string, testing for, 320
End-of-file marker, 123, 165
Envelope, disk, 13-14
Environment, DOS. *See* DOS environment
ERASE command, 620-622
Erasing files, 78-80
ESC key
canceling a command with, 152-153
location of, 153
Escape character
how to enter, 261-262
using PROMPT command to write, 465
Escape sequences. *See* ANSI escape sequences
EXE2BIN command, 622-624
EXIT command, 470-472, 624-625
Extended character set, 174-176
enhancing display of, 475-477
using in subdirectory names, 387-388
using to protect your files, 387-388
with PROMPT, 466
Extensions, file-name. *See* File extensions
External DOS commands, 40-41, 43-44
accessing, 47
defining a path to, 277-278
in distinct subdirectories, 382
residing in subdirectories, 364-365
External files, locating, 376-382
External floppy-disk drives
supporting, 254-256
system with, 255

F

F1, copying characters to the command line, 166, 167
F2, copying characters to the command line, 170
F3, copying commands to the command line, 167-168
F4, deleting characters on the command line, 171
F5, editing current command line, 171-172
F6, placing end-of-file marker, 122-123, 165, 200
Fan, computer, 10
FASTOPEN command, 468-470, 625-628

FC command, 409, 425-430, 628-634
/A qualifier, 428
/C qualifier, 429
/N qualifier, 428
qualifiers to, 426
FCBS= entry in CONFIG.SYS, 234, 246
FDISK.COM file, 101
FDISK command, 101-104, 635-642
File attributes (*See also* ATTRIB command)
setting and displaying, 416-422
File control block (FCB), 245
File extensions, 56, 62
and file types, 58-59
DOS files, 120
included in DIR command, 63
omitting in target file names, 72
to group files by function, 119
File handles, 245, 247
File-manipulation commands, 409-435
File names, 56, 58
and file types, 59
changing, 75-78
displaying, 107-108
included in DIR command, 62-63
meaningful, 65-66, 126
valid, 65
wildcard characters in, 66-69
File replacements, selective, 430-435
File sharing, in local-area networks, 477-479
File size, 56
displayed in bytes, 59
File size, *continued*
suppressing in directory display, 60, 62
Files
backing up, 503-526
combining, 440-441
comparing two, 422-430
copying, 439-455
copying in subdirectories, 363
copying multiple, 72-74
deleting, 78-80
deleting multiple, 79-80
dispersed on disk, 111-112
displaying in subdirectories, 374-376
displaying names of, 56
enhancing with ALT, 172-178
fragmented, 522-524
making duplicate copies of, 70-75
maximum number on a disk, 339-340
naming, 65-66
number of in one directory, 75
number of that DOS can open, 247-248
printing, 132
printing multiple, 215-217
protecting with unique subdirectory names, 387-388
recovering damaged, 412-416
removing in subdirectories, 352
setting to read-only, 416-421
to store information, 55
FILES= entry in CONFIG.SYS, 235, 248
FIND command, 196, 409, 642-645
/C qualifier, 198-199

FIND command, *continued*
for displaying selected output, 191
/N qualifier, 197
searching multiple files, 410-412
/V qualifier, 198
Fixed-disk system, 4
Fixed disks (*See also* Disks)
backing up to floppies, 506
installing DOS on, 382-384
formatting, 104-105
preparing, 100-105
versus floppy disks, 100-101
Floppy-disk segments, 109
Floppy disks (*See also* Disks)
backing up, 504
bending, 17
formatting, 86, 88-95
illustrated, 12
inserting in drives, 17-19
versus fixed disks, 100-101
FOR command, 321-324, 645-647
and wildcard characters, 323
from the DOS prompt, 324
named parameters with, 330
FORMAT command, 86, 88-95, 647-657
exit status values, 314
/4 qualifier, 95
/S qualifier, 92-94, 105, 508
Formats, for country-specific symbols, 490
Formatted disks, backing up to, 508-509
Formatting fixed disks, 104-105
Formatting floppy disks, 86, 88-95
Fragmentation, disk, 522-524
Fragmented file, 522-524
Function keys (*See also* F1 through F6)
location of, 166
predefined, 680-681
using, 165-168

G

GOTO command, 657-658
for batch-file branching, 315-320
to skip or repeat commands, 316
used with IF command, 319
GRAFTABL command, 475-477, 658-661
/STA qualifier, 477
Graphics, printing, 162-164
GRAPHICS command, 162-164, 661-663
/R qualifier, 164

H

Handling disks, 86-87
Hardcopy listings, 211
Hardware
defined, 11
standard components, 7
Hardware devices
additional, 119-120
standard, 119-120
Hardware interrupt stack, creating, 251-252
Heat and sunlight, cautions, 16-17
Hidden files, DOS, 107
and the REPLACE command, 435
Home position, 26

I

IBM PC keyboard layout, 147-148
IBM PC/AT keyboard layout, 147-148
IBM ProPrinter, 493
IBM QuietWriter printer, 493-494
IF command, 663-666
    to make decisions, 312-321
    used with GOTO command, 319
IF ERRORLEVEL command, 314-315, 536
    and XCOPY, 454
IF EXIST FILE command, 312-313
IF string1==string2 command, 313-314
Input, keyboard, 247
Input/output redirection, 183-207
Input redirection operator (<), 189-190
INS key
    location of, 169
    used on command line, 169-170
Insert mode, 170
Installing DOS on fixed disk, 382-384
Interactive commands, 287
Internal DOS commands, 40-44
International DOS support, 485-499
Interrupt stack, creating, 251-252
I/O redirection, 183-207
    DIR command using default, 185
    keyboard to screen display, 184
    software supporting, 207
I/O redirection operators
    < (input), 189-190
I/O redirection operators, *continued*
    > (output), 183-186
    >> (append), 187-189
    using multiple, 206-207

J

JOIN command, 399-402, 666-669
    /D qualifier, 401-402

K

Key combinations, entering, 149
KEYB command, 669-673
    format of, 487-488
    in AUTOEXEC.BAT, 492
Keyboard connection, hardware, 11
Keyboard layouts, IBM PC and PC/AT, 147-148
Keyboard templates, selecting, 486-489
KEYBOARD.SYS device driver, 254, 488
Keys, reassigning
    ANSI escape sequences for, 266-267
    batch file for, 306-307

L

LABEL command, 97-99, 107, 673-677
    for naming a fixed disk, 105
Label names
    batch-file, 316-319
    disk, 29, 515-516
LANs, 477-479
Laser printers, connecting, 141

LASTDRIVE= entry in
CONFIG.SYS, 235, 248-249, 404-405
Line numbers, displaying, 197
Local-area networks, 477-479
Log files, 520-521
Logical disk drives
defining the number of, 248-249
supported, 404-405
using, 393-405
LPT1 port, 130-131, 161-162
as PRN device, 212

M

Magnetic and electronic devices, cautions, 14-15
MCOPY command, creating, 453
MD command. *See* MKDIR command
Memory, reading DOS into, 20
Memory in use/memory available display, 107
Memory-resident software, and the DOS environment, 280-282
Messages, using batch files to display, 296-297
Metacharacters, 464
Microfloppy disks, 19, 28
MKDIR command, 341-346, 353-357, 677-680
MODE command, 133-138, 680-689
abbreviations of qualifiers, 498-499
format of, 140-141
parameters to, 140-141
REFRESH qualifier, 497
/STATUS qualifier, 498
MODE command, *continued*
table of values, 136
to align text on the screen, 137-138
to change output format, 139-140
to connect two screens, 135-136
to use COM1, 161
with codepages, 495-499
Modem, 131
Monitor connection, hardware, 10
MORE command, 689-691
for displaying output, 190-191
output suspended by, 192
redirecting output of, 194
resumption of output with, 193
use of pipe with, 202
without redirection, 199, 201
MS-DOS, and PC-DOS, 25
Multiple files
deleting, 79-80
searching with FIND, 410-412
Multiple-language support, 492-495

N

Named batch parameters, 328-331
Naming directories, 344-345, 358
Naming disks, 95-100
Naming files, 65-66
Nested batch files, 331-333
NLSFUNC command, 494, 691-692
NOT operator, in IF commands, 315
Null string, 320

O

Operating systems, 12

Operating systems, *continued*
using more than one, 103
Operators. *See* I/O redirection operators
Output (*See also* I/O redirection)
screen, 247
suspending with CTRL-S, 156-159
suspending with DIR /P, 156-159
Output device, defining to PRINT, 224-225
Output redirection operator (>), 184-186
Overstrike mode, 170

P

Parallel connector, 130
Parallel data communication, 128-133
Parallel port. *See* LPT1 port
Parallel printers, 128-133
character mode transmission, 243-244
output to, 161
Parameters, in command formats, 244
Parity, setting, 140
Partitions, DOS, 102-104
PATH command, 277-278, 376-382, 693-694
in AUTOEXEC.BAT file, 382
PATH= entry in CONFIG.SYS, 277-278, 376-377
PAUSE command, 694-696
to suspend batch processing, 300-301
PC-DOS, and MS-DOS, 25, 39
Periods, in file names, 58
Pipe character (|), location of key, 201
Pipe operator
and disk space, 205
and write protection, 203
redirecting output with, 201-203
used with DIR command, 203
used with MORE command, 202
Ports, printer. *See* Printer ports
Predefined function keys, 696-697 (*See also* F1 through F6)
Pretender commands, 393-405
Print buffer
increasing the size of, 226-227
specifying the size of, 227
PRINT command, 211-229, 697-703
and invalid device names, 213-214
and nontext files, 217
and task switching, 218-219
/B qualifier, 226-227
/C qualifier, 219-220, 222-223
configuration qualifiers, 223-229
/D qualifier, 224-225
how it works, 218-219
increasing processing time of, 227-229
/M qualifier, 227-228
order of printing, 216-217
output to the PRN device, 212-213
/P qualifier, 222-223
printing multiple files with, 215
/Q qualifier, 225

PRINT command, *continued*
/S qualifier, 228
specifying buffer size with, 227
/T qualifier, 222
/U qualifier, 228-229
use of the buffer by, 226
wildcard characters with, 217
Print queue, 215
adding files to, 222-223
canceling, 222
deleting a file in, 217-218
increasing the size of, 225
listing the files in, 216-217
memory consumed by, 226
removing files from, 219-223
Printer output
controlling, 138-140
8 lines per inch, 139
132 characters per line, 139
Printer ports, 130 (*See also* AUX port; COM1 port; LPT1 port; PRN port)
parallel, 131
serial, 131
PRINTER.SYS device driver, 254, 493-494
Printers
connecting laser, 141
determining the type, 129-130
output redirected to, 185
parallel, 128-133
serial, 128-133
Printing
files, 132
graphics, 162-164
multiple files, 215-217
the screen display, 160-164
ways of, 211
PRN port, 131, 212-213
Processor speed, 147
Program Disk, 12-13
file types on, 58
Programs, creating with DEBUG, 532-543
PROMPT command, 462-467, 703-705
in AUTOEXEC.BAT, 466-467
metacharacters supported, 463-464
with ANSI.SYS, 465
Prompts, screen
changing the system, 279-280
customizing the system, 462-467
DEBUG, 533
DOS, 25, 37-38, 47
Protecting files, 387-388
PS/2 models, 5

Q

Quad-density disks, 94

R

RAM disks. *See* Virtual disks
RAMDRIVE.SYS device driver, 260
format of, 257
parameters to, 257-258
Random access memory (RAM), 257
Read-only files, 416-421
and the DEL command, 79
and the RENAME command, 77
Reading parts of a file, 110-111
RECOVER command, 409, 412-416, 705-707
Redirecting I/O. *See* I/O redirection

REM command, 298-300, 708
Remarks, in batch files, 298-300
Removing files in subdirectories, 352
Removing subdirectories, 350-353
REN command. *See* RENAME command
RENAME command, 75, 709-710
  and moving files, 76
  and read-only files, 77
  and wildcard characters, 78
  and write-protected disks, 77-78
  with subdirectories, 363-364
REPLACE command, 410, 430-435, 710-716
  /A qualifier, 434
  exit status codes, 435
  format of, 431
  /S qualifier, 434
Resources, computer, 12
Restarting DOS, 31
  cautions, 150
  with CTRL-ALT-DEL, 149
RESTORE command, 506-507, 515, 716-722
  exit status values, 522
  format of, 513
  parameters to, 514
Restoring files
  to fixed disk, 513-515
  to fixed disk from floppies, 506-507
Retrying a command, 45-46
RMDIR command, 350-353, 723-724
Root directory
  \ to represent, 342
  as current directory, 347
Root directory, *continued*
  maximum subdirectories in, 340
Round robin task switching, 218-219
Runaway programs, 153

S

Screen, clearing, 262-263
Screen attributes
  setting with ANSI escape sequences, 263-266
  setting with batch files, 297-298, 301-302, 306
Screen color
  changing with ANSI escape sequences, 263-265, 303
  changing with batch files, 297-298, 301-302, 306
Screen display
  changing to 40-column, 134-135
  controlling, 134-138
  default 80-column, 134
  printing, 160-164
Screen messages, creating, 265-266
Screens, connecting monochrome and color, 135-136
Scrolling, of directory listings, 60
Search path, for data files, 384-387
Secondary command processor, terminating, 472
Sectors, disk, 239-240, 413-414
  consecutive, 110
  damaged, 90, 105-107
Segments, floppy disk, 109
SELECT command, 499, 724-729
Serial connector, 130

Serial data communications, 128-133
Serial port. *See* COM1 port
Serial printers, 128-133
  output to, 161-162
SET command, 275-283, 378, 462-463, 729-731
SHARE command, 477-479, 731-733
  parameters to, 477-478
SHELL= entry in CONFIG.SYS, 235, 250-251
  and COMSPEC= in DOS environment, 277
  to define DOS environment size, 282
SHIFT command, 324-328, 733-734
  used with IF and GOTO commands, 326
SHIFT-PRTSC
  and graphics, 162
  location of keys, 161
  printing with, 160-164
  with text, 162
Single-disk drive system
  copying disks, 30
  using drives A and B in, 394-397
Size of files display, 56
Software
  defined, 11
  updating, 430-435
SORT command, 194, 735-739
  and write protection, 205
  for displaying sorted output, 190
  /+n qualifier, 196
  /R qualifier, 195
  redirected output with, 205
  used with DIR, 204
SORT command, *continued*
  without redirection, 199-200
Source disk, 31
Space character, in batch-file parameters, 304
Speed of data communication, 133
Stack, memory, 251
STACKS= entry in CONFIG.SYS, 235, 251-252
Starting DOS, 3
Starting your system, 8-31
Stdin (standard input), keyboard as, 247
Stdout (standard output), screen as, 247
Stop bits, setting, 141
Storage media, 14-15
Subdirectories (*See also* Directories)
  advanced use of, 371-390
  creating, 341-346
  creating multiple levels of, 353-354
  external commands in, 364-366, 382
  maximum in root directory, 340
  number on a disk, 341
  removing, 350-353
  removing files in, 352
  using, 339-366
Subdirectory files
  directory listings of, 361-362
  manipulating, 362-364
Subdirectory names
  abbreviating, 402-405
  to protect files, 387-389
SUBST pretender command, 399, 402-405, 739-741
  /D qualifier, 404

SUBST command, *continued*
disabling, 249
Supplemental Programs Disk, directory of, 63-64
Surge suppressor, 7-8
Symbol sets, 490-492
SYS command, 389-390, 525, 742-744
SYS extension, 253
System commands, 459-479
System date
accuracy of, 59-60
changing, 49
displaying, 48
entering, 20-21
setting, 48
suppressing in directory display, 60, 62
using the correct, 24
System files, and disk space, 93-94
System performance, improving, 238-242
System prompt, 25, 37-38
changing, 279-280
containing default disk drive, 47
customizing, 462-467
System services, DOS, 543
System startup, 233-234
System time
accuracy of, 59-60
displaying, 50
setting, 50

T

Target disk, 31
not enough space on, 74-75
Target file
assigning a different name to, 71
existing file name as, 76
Target file names
and source file names, 71-72
and wildcard characters, 72
omitting the extension, 72
Task switching, 218-219
Tax information, tracking, 96-99
Telephones, and disks, 15
Templates, keyboard, 486-489
Terminating a command, 44-46
with CTRL-BREAK, 153, 156
with CTRL-C, 154-156
Text files
creating, 119-126
definition, 121
displaying, 124-125
Time
entering the current, 22
in directory listings, 59
on PC/AT, 22, 24
setting the current, 50
standard versus military, 22-23
using the correct, 24
TIME command, 50, 745-746
Time prompt, 22
TREE command, 373-375, 747-750
/F qualifier, 374-375
TTY device, 467
Two-floppy disk system, 3-4
TYPE command, 108, 750-752
error message with, 125
in batch files, 321-323
to display text files, 124-125
with subdirectories, 362-363

U

Updating software, 430-435
Upgrading DOS, 389-390, 525-526
User response, programs to determine, 540-542

V

VDISK.SYS device driver, 258
VER command, 39-40, 42-44, 752
VERIFY command, 752-754
Verifying disk output, 459-467
Verifying file copies, 443-444
Versions of DOS, 40-41
    determining which you are using, 39
Virtual disks
    accessing, 258-259
    characteristics of, 257
    command files on, 260
    installing, 257-260
    uses for, 259
VOL command, 99-100, 754-755

W

Wildcard characters (? and *)
    and the FC command, 429
    and the FOR command, 323
    and the RENAME command, 78
    for copying files in subdirectories, 363
Wildcard characters, *continued*
    in target file names, 72
    to copy multiple files, 72-74
    with ATTRIB, 418
    with COMP command, 425
    with DIR, 66-69
    with PRINT command, 217
    with PRINT /C command, 221
Write-protect notch, 26
Write-protect opening (microfloppy disks), 28
Write-protect tabs (floppy disks), 27
Write-protected disks
    and the DEL command, 79
    and the RENAME command, 77-78
Write protection
    and pipe operator, 203
    and SORT command, 205
    of original disks, 26-27

X

XCOPY command, 445-455, 755-761
    /E qualifier, 449
    exit status codes, 454
    format of, 445
    /M qualifier, 451-452
    /P qualifier, 454-455
    parameters to, 445-446
    /S qualifier, 449
    versus COPY command, 454-455
    with ATTRIB command, 450

IF YOU ENJOYED THIS BOOK...

help us stay in touch with your needs and interests by filling out and returning the survey card below. Your opinions are important, and will help us to continue to publish the kinds of books you need, when you need them.

What brand of computer(s) do you own or use? _____

Where do you use your computer the most? ☐ At work ☐ At school ☐ At home

What topics would you like to see covered in future books by Osborne/McGraw-Hill?

_____

How many other computer books do you own? _____

Why did you choose this book?

- ☐ Best coverage of the subject.
- ☐ Recognized the author from previous work.
- ☐ Liked the price.
- ☐ Other

Where did you hear about this book?

- ☐ Book review.
- ☐ Osborne catalog.
- ☐ Advertisement in: _____
- ☐ Found by browsing in store.
- ☐ Found/recommended in library
- ☐ Other

Where did you find this book?

- ☐ Bookstore
- ☐ Computer/software store
- ☐ Department store
- ☐ Advertisement
- ☐ Catalog
- ☐ Required textbook
- ☐ Library
- ☐ Gift
- ☐ Other

Where should we send your FREE catalog?

NAME _____

ADDRESS _____

471-5 CITY _____ STATE _____ ZIP _____

DOS

SECRETS, SOLUTIONS, SHORTCUTS

DOS: Secrets, Solutions, Shortcuts is filled with hundreds of tips and so many practical ideas that every DOS user, from novice to pro, can use it immediately to get more from MS-DOS® and PC-DOS.

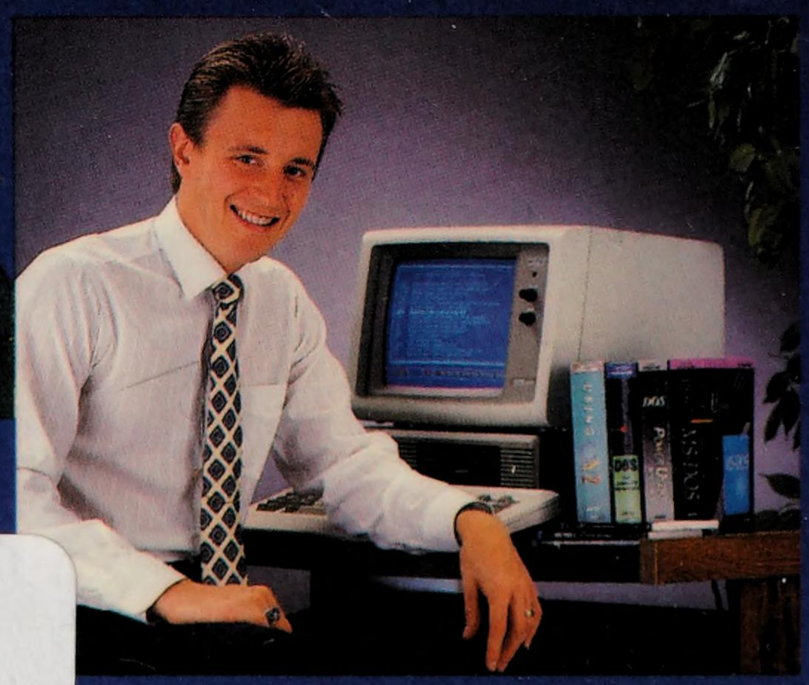
Kris Jamsa, the author of the acclaimed **DOS: The Complete Reference, Using MS-DOS®, DOS: Power User's Guide,** and **DOS: The Pocket Reference**, now shares his own collection of DOS shortcuts and the powerful techniques which he has accumulated since the release of DOS in 1981. You'll find techniques in this book that have never before been available.

Each chapter examines a specific DOS concept which you'll use on a daily basis. Jamsa explains the concept in detail, presenting tips that will make you more productive and pitfalls to avoid.

DOS: Secrets, Solutions, Shortcuts makes every DOS user even better with DOS. Learn the techniques it took the expert years to develop.

Kris Jamsa holds a B.S. degree in computer science from the United States Air Force Academy and an M.S. degree in computer science from the University of Nevada at Las Vegas. He is also the author of **DOS: The Complete Reference, Using MS-DOS®, DOS: Power User's Guide, DOS: The Pocket Reference, Using OS/2™, OS/2™: The Pocket Reference, Turbo Pascal® 4: The Pocket Reference, Turbo Pascal® Programmer's Library, Turbo C® Programmer's Library,** and **The C Library.**

TRADEMARKS: MS-DOS is a registered trademark of Microsoft Corp. OS/2 is a trademark of International Business Machines, Corp. Turbo C and Turbo Pascal are registered trademarks of Borland International.



0078814715
02/20/2017 14:59-2
22

$24.95